



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

D.A.N.G.E.R - Disaster Prevention. Videojuego educativo
para la gestión de emergencias en Unity 3D

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Álvarez Zavaleta, Luis Alberto

Tutor/a: Mollá Vayá, Ramón Pascual

Cotutor/a: Pellicer Armiñana, Teresa María

CURSO ACADÉMICO: 2021/2022

Resumen

Videjuego de simulación social y estrategia en Unity 3D. El proyecto parte del trabajo realizado por tres alumnos en cursos anteriores.

Actualmente el videjuego cuenta con tres modos de juegos diseñados con el objetivo de crear un entorno interactivo, seguro y divertido que facilite aprender medidas y comportamientos que deben tomarse en una situación de incendio.

Este trabajo consistirá en analizar las posibles mejoras que se puedan implementar en los diferentes niveles diseñados. Este proyecto abarcará mejoras centradas en la percepción del entorno y los patrones de comportamiento de la inteligencia artificial, la generación procedural de niveles y el diseño de nuevas mecánicas en los distintos niveles existentes.

Se mantendrá actualizado el Documento de diseño del videjuego (GDD) con los cambios y las modificaciones introducidas en los distintos niveles para futuros trabajos y expansiones de este.

Palabras clave: Simulación social, Unity 3D, IA, generación procedural, videjuegos

Abstract

Social simulation and strategy videogame in Unity 3D. The project is based on the work done by three students in previous courses.

Currently the game has different levels designed with the aim of creating an interactive, safe, and fun environment that facilitates learning measures and behaviors to be taken in a fire situation.

This work will consist of analyzing the possible improvements that can be implemented in the various levels designed. This project will cover improvements focused on the perception of the environment and the behavior patterns of artificial intelligence, the procedural generation of levels and the design of new mechanics in the different existing levels.

The Game Design Document (GDD) will be kept updated with the changes and modifications introduced in the different levels for future work and expansions of the game.

Keywords: Social simulation, Unity 3D, AI, procedural generation, video games.



Resum

Videojoc de simulació social i estratègia en Unity 3D. El projecte parteix del treball realitzat per tres alumnes en cursos anteriors.

Actualment el videojoc compta amb diferents nivells dissenyats amb l'objectiu de crear un entorn interactiu, segur i divertit que facilite aprendre mesures i comportaments que han de prendre's en una situació d'incendi.

Aquest treball consistirà a analitzar les possibles millores que es puguin implementar en els diferents nivells dissenyats. Aquest projecte abastarà millores centrades en la percepció de l'entorn i els patrons de comportament de la intel·ligència artificial, la generació procedural de nivells i el disseny de noves mecàniques en els diferents nivells existents.

Es mantindrà actualitzat el Document de disseny del videojoc (GDD) amb els canvis i les modificacions introduïdes en els diferents nivells per a futurs treballs i expansions d'aquest.

Paraules Clau: Simulació social, Unity 3D, IA, generació procedural, videojocs.



Tabla de contenido

1.	Introducción	9
1.1	Motivación.....	9
1.2	Objetivos.....	10
2.	Estado del arte.....	11
2.1	Ludificación.....	11
2.2	Aprendizaje Basado en Juegos	14
2.3	Impacto de los videojuegos	16
3.	Herramientas	18
3.1	Introducción.....	18
3.2	Motor de Videojuego.....	18
3.3	Visual Studio (Entorno de desarrollo integrado).....	21
3.4	GitHub (Controlador de versiones)	22
3.5	Angular (Framework para la página web).....	22
3.6	Firebase (Base de datos)	23
4.	Análisis	24
4.1	Introducción.....	24
4.1.1	Modo Construcción	24
4.1.2	Modo Escape	25
4.1.3	Modo Aventura:	25
4.1.4	Página web	26
4.2	Propuestas de mejoras	27
4.2.1	Modo Construcción:.....	27
4.2.2	Modo Escape: Mejorar Generación Procedural	28
4.2.3	Modo Escape: Inteligencia Artificial	28
4.2.4	Modo Aventura	30
4.2.5	Página web: Cuestionario.....	31
5.	Implementación	32
5.1	Introducción.....	32
5.2	Modo Construcción	32
5.2.1	Mejoras en la interacción con el usuario.....	33
5.2.2	Sistema de Guardado.....	33
5.3	Modo Escape	35
5.3.1	Animaciones.....	35
5.3.2	Sensores.....	39



5.3.3	Inteligencia artificial	42
5.3.4	Semilla Aleatoria.....	44
5.3.5	Generación Procedural	44
5.3.6	Generación de fuego.....	46
5.4	Modo Aventura.....	47
5.5	Página web.....	49
5.5.1	Cuestionario	49
5.5.2	Catálogo de habitaciones.....	50
6.	Conclusiones	52
6.1	Comparación con la última versión(changelog).....	52
6.1.1	Relación con los estudios cursados	52
6.1.2	Trabajos futuros.....	53
7.	Agradecimientos	54
8.	Bibliografía	55
	Anexo I: Glosario	57
	Anexo II: ODS.....	58
	Anexo III: Documento del diseño del juego	60
	SECCIÓN 1: DATOS PRINCIPALES	60
	SECCION 2: Pitch doc	61
	SECCIÓN 3: Análisis competitivo	63
	SECCIÓN 4: Estructura narrativa, personajes.....	67
	SECCIÓN 5: Escenas, Niveles	68
	SECCIÓN 6 Assets:.....	74
	SECCIÓN 7: Controles.....	83
	SECCIÓN 8: Interfaz.....	85
	SECCIÓN 9: Herramientas complementarias	91

Table de Figuras

Figura 1. Intervenciones por incendio o explosión(Fundación MAPFRE y APTB 2021).....	9
Figura 2. Interfaz de Kahoot para crear un cuestionario	13
Figura 3. Aplicación móvil Habitica	13
Figura 4. Aplicación móvil Nike Running	13
Figura 5. Makeblock mbot Neo.....	14
Figura 6. Dragonbox.....	15
Figura 7. Cities Skylines	15
Figura 8. Minecraft Education Edition.....	16
Figura 9. Civilization IV	16
Figura 10. Número de jugadores por región en el mundo.....	17
Figura 11. Tiempo dedicado a videojuegos (Asociación Española de Videojuegos 2021).....	17
Figura 12. Gráfico de porcentaje de videojuegos creados con distintos motores ...	18
Figura 13. Interfaz de usuario de Unity Engine	19
Figura 14. Interfaz de usuario de Unreal Engine.....	20
Figura 15. Interfaz de usuario de Cryengine	20
Figura 16. Visual Studio.....	22
Figura 17. Interfaz de Github	22
Figura 18. Interfaz de la consola de Firebase	23
Figura 19. Menú de inicio del videojuego DANGER.....	24
Figura 20. Modo Construcción.....	25
Figura 21. Modo Escape.....	25
Figura 22. Modo Aventura	26
Figura 23. Portal de la Página web.....	26
Figura 24. Assets de oficina	27
Figura 25. Nivel generado en el Modo Escape con poca variedad	28
Figura 26. El NPC no es capaz de detectar el fuego que está en su espalda	29
Figura 27. Árbol de comportamiento del personaje	30
Figura 28. Esquema del nuevo nivel	30
Figura 29. Nuevo Menú principal	32
Figura 30. Ejemplo de habitación creada	32
Figura 31. Previsualización de una silla	33
Figura 32. Estructura Del Gameobject Room	33
Figura 33. Interfaz de Guardado de Partida	34
Figura 34. Partículas del extintor activadas.....	35
Figura 35. BoxCollider de la partícula de fuego (izquierda) y configuración de la partícula de fuego (derecha)	35
Figura 36. Puerta cerrada (izquierda) y Puerta abierta (derecha).....	36
Figura 37. Personaje importado a Mixamo probando una animación de caminar ..	36
Figura 38. Componente animator de un personaje.....	37
Figura 39. Árbol de Mezcla del estado Movimiento.....	37
Figura 40. Avatar del personaje (izquierda) y Máscara del avatar para los brazos (derecha)	38



Figura 41. Animación de extintor(derecha) y configuración de la Capa Extintor en el componente Animator	38
Figura 42. Personaje en espera agachado(izquierda) y Personaje moviéndose(agachado)	39
Figura 43. Personaje Asustado(derecha) y personaje muerto (izquierda).....	39
Figura 44. Etiqueta y capa del Gamobject fuego(izquierda) y Capas de detección del script Sensor	40
Figura 45. Ejemplo de detecciones implementadas por el sensor en caso de detectar fuego	41
Figura 46. Arbol de comportamiento implementado para los NPC del modo Escape	42
Figura 47. Cálculo de distancias usando el Navmesh generado por Unity	43
Figura 48. Interfaz de creación en el modo Escape.....	44
Figura 49. Prefab de una de las habitaciones	45
Figura 50. Nuevo generado procedural	46
Figura 51. Personaje Apagando casillas del fuego(izquierda) y propagación del fuego unos segundos despues(derecha)	47
Figura 52. Detección de sensores en el modo Aventura	47
Figura 53. Árbol de comportamiento de los NPC en el modo Adventure	48
Figura 54. Nuevo nivel Modo aventura.....	48
Figura 55. Nueva interfaz del cuestionario	49
Figura 56. Interfaz de usuario.....	50
Figura 57. Interfaz para subir un juego a Firebase.	51
Figura 58. Apartado de Catálogo de la página web	51

1. Introducción

1.1 Motivación

La motivación personal que me llevó a continuar el desarrollo de este proyecto es mi pasión por los videojuegos. Desde pequeño los videojuegos se convirtieron en una de mis principales formas de entretenimiento junto a las películas. Siento que los videojuegos son una forma de aprendizaje atractiva e interesante y puede ser usado por los docentes en el proceso de enseñanza-aprendizaje de los alumnos.

Según un estudio de la Asociación Profesional de Técnicos de Bomberos (APTB) y la fundación de Mutualidad de la Agrupación de Propietarios de Fincas Rústicas de España (Mapfre), durante el año 2020, se han computado un total de 123.800 siniestros por incendio y se registraron 164 muertos (Fundación MAPFRE y APTB 2021).



Figura 1. Intervenciones por incendio o explosión(Fundación MAPFRE y APTB 2021)

Debido a este número tan elevado de casos de incendio y muertes, surgió el interés por el desarrollo de un videojuego didáctico que enseñe a los jugadores de las acciones que se deben realizar en caso de incendios, con el objetivo de ayudar a reducir el número de incendios y muertes.

Considero que el desarrollo de este videojuego presenta un estilo interesante y atractivo que otros al igual que yo, podrán utilizarlo para disfrutar y aprender de forma didáctica sobre planes de prevención de incendios, combinando elementos de los géneros de Estrategia en Tiempo Real (RTS) en los cuales se debe gestionar, trazando un plan y priorizando diferentes estrategias para cada situación con la generación procedural de niveles para diversificar los escenarios en el juego

1.2 Objetivos

El objetivo principal que se pretende conseguir es

- Desarrollar un videojuego, que permita crear un entorno donde los jugadores puedan adquirir conocimiento sobre las normas básicas de actuación en situaciones de emergencia.

Entre los objetivos específicos necesarios para cumplir el objetivo principal se encuentran:

- Implementar nuevos sistemas de interacción dentro del videojuego, así como diseñar nuevos niveles, con el objetivo de mejorar la experiencia que perciben los jugadores para hacerla más interesante y atractiva
- Captar el interés de los jugadores, permitiéndoles interactuar de forma continua con los diferentes niveles, de manera que el jugador se mantenga motivado a aprender y experimentar las diferentes mecánicas del videojuego
- Diseñar una curva de aprendizaje rápida para que los jugadores comprendan cómo se pueden aplicar los nuevos conocimientos adquiridos a situaciones reales
- Desarrollar el videojuego con un código fuente legible y documentado para que este sea flexible, escalable y fácil de mantener y modificar.
- Elaborar y mantener un documento de diseño del videojuego (GDD) claro y conciso para que el videojuego pueda seguir desarrollándose.

Además de intentar cumplir estos objetivos, también se pretende mejorar la página web desarrollada, con el objetivo de fomentar y expandir los distintos métodos de aprendizaje existentes y así ofrecer al público una variedad de herramientas didácticas para fomentar al jugador en la búsqueda de nuevas experiencias de aprendizaje sobre las medidas necesarias a tomar en caso de incendio.

2. Estado del arte

Según la definición proporcionada por Chris Crawford, “un juego es un sistema formal cerrado que representa subjetivamente un subconjunto de la realidad” (Crawford et al. 2003, p. 9).

Otra definición por parte de Huizinga define al juego como:

[..] una acción u ocupación libre, que se desarrolla dentro de unos límites temporales y espaciales determinados, según reglas absolutamente obligatorias, aunque libremente aceptadas, acción que tiene su fin en sí misma y va acompañada de un sentimiento de tensión y alegría y de la conciencia de “ser de otro modo” en la vida corriente.(Huizinga 2012)

A partir de estas definiciones se pueden obtener algunas características que definen a un juego

- Es una forma que permite experimentar diferentes emociones y situaciones mientras se excluyen de los peligros y las consecuencias que estas supondrían en la realidad.
- Debe estar definido dentro de un marco espacio temporal marcado o pactado con anticipación
- Sus reglas deben estar definidas y presentar al jugador distintas opciones con las que conseguir un objetivo claro y conciso, se trata de presentar al jugador con un número amplio de opciones con las que conseguir este objetivo
- Es una actividad libre, voluntaria y creativa con distintos fines, los cuales pueden ir desde fines educativos hasta fines lúdicos o con la intención de desarrollar habilidades o destrezas.

Los juegos no solo toman un papel de entretenimiento, también son de gran interés en la educación, la formación y el crecimiento de los niños, (UNICEF 2018) las clasifica como una de las formas más importantes en las que los niños pequeños obtienen conocimientos y competencias esenciales, además lo consideran como una herramienta esencial para lograr una enseñanza primaria universal y alcanzar los Objetivos de Desarrollo Sostenible (ODS). Todo ello orientado en el aprendizaje preescolar basado técnicas lúdicas que permita mejorar y fortalecer el conocimiento de una forma alternativa a los métodos tradicionales.

Por lo tanto, es de especial interés analizar las ventajas de la incorporación de los juegos al ámbito académico, a fin de poder mejorar los procesos de enseñanza y aprendizaje en todos los niveles educativos.

2.1 Ludificación

La ludificación o *gamificación* tiene por objeto introducir estructuras creativas e innovadoras provenientes de los juegos para convertir una actividad, *a priori* aburrida, en otra actividad que motive a la persona a participar en (Romero Sandí y Rojas Ramírez 2013).



En otras palabras, la ludificación es una metodología de enseñanza que utiliza la dinámica de los juegos para motivar a los estudiantes, permitiendo mejorar sus habilidades, destrezas y aptitudes de aprendizaje.

Diversos autores han sistematizado los pasos a seguir para diseñar un planteamiento ludificado. Algunos de ellos, como (Pérez-López 2018), habla de 6 momentos claves (lo que lleva a jugar, lo que pone a jugar, lo que hace avanzar, lo que mantiene en el juego, lo que enriquece la experiencia y lo que permite jugar) y 15 elementos de *gamificación* (temática, narrativa, objetivo, misiones, capacidad de decisión, puntos, *feedback*, *flow*, vidas, interacción, misiones extras, premios, avatar, reglas y curiosidad).

Las etapas para realizar ambientes de aprendizaje gamificados se caracterizan por analizar el contexto de la población objetivo; identificar los objetivos y rutas de aprendizaje; establecer las tareas individuales y colectivas; definir las recompensas a otorgar y brindar la posibilidad de repetir actividades (Ardila-Muñoz 2019).

Por lo tanto, para conseguir un buen diseño de un planteamiento ludificado (Yu-kai Chou 2019) plantea la metodología Octalysis. Se basa en la integración de los ocho elementos de la motivación, en el proceso de gamificación:

- Sentido y vocación épicas: La ludificación debe hacer sentir al jugador que está actuando por una causa mayor, por ejemplo, el hecho de participar en un programa de reciclaje hace sentir al jugador que contribuye a reducir el calentamiento global.
- Desarrollo y realización: Los jugadores deben tener una sensación de progreso y logro mientras juegan. Deben sentir que están mejorando y logrando algo.
- Potenciación de la creatividad y retroalimentación: Los jugadores deben sentir que tienen libertad para ser creativos y que sus esfuerzos son reconocidos y recompensados, proponiendo nuevos métodos que alimenten el contenido del producto.
- Propiedad y posesión: Los jugadores deben tener un sentimiento de propiedad sobre sus personajes y objetos de juego. Deben sentir que pueden tomar decisiones que afectan al mundo del juego.
- Influencia social: Los jugadores deben sentir que sus interacciones sociales dentro del juego importan y que tienen la capacidad de formar parte de una comunidad.
- Escasez e impaciencia: Los jugadores deben sentir una sensación de urgencia y deseo por el contenido del juego. Deben sentir que necesitan progresar para obtener mejores recompensas.
- Imprevisibilidad y curiosidad: el producto debe transmitir una sensación de sorpresa, siendo difícil prever el resultado de una acción generando intriga y motivando al jugador a continuar hasta descubrirlo.
- Pérdida y evasión: Los jugadores deben tener una sensación de pérdida cuando cometen errores, sintiendo la necesidad de evitar perder para poder seguir jugando.

A partir de estos ocho drivers se puede detectar cuáles son los más estimulantes para nuestro público, analizar los puntos fuertes e introducir nuevas mecánicas de juego apropiadas para de nuestro producto.

Algunos ejemplos de herramientas creadas a partir de esta metodología son las siguientes

- Kahoot¹: Herramienta utilizada para crear y diseñar juego de trivia en tiempo real, se utilizan en los entornos educativos como una herramienta para ayudar a los estudiantes a aprender de manera divertida y entretenida.

¹ <https://kahoot.com/>

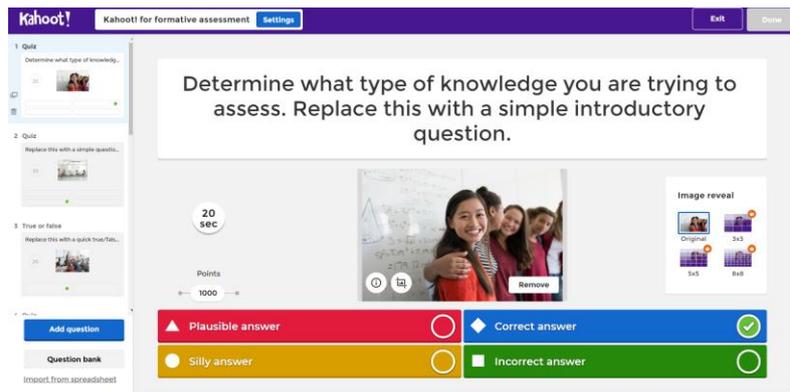


Figura 2. Interfaz de Kahoot para crear un cuestionario

- Habituca²: Se trata de una aplicación para móvil creada con el objetivo de mejorar la productividad. Está desarrollada como un juego de rol que permite rastrear tus hábitos y objetivos y otorga recompensas según los vayas completando



Figura 3. Aplicación móvil Habituca

- Nike Running App³: La aplicación se diseñó para ayudar a los corredores a mejorar su rendimiento y alcanzar sus objetivos de carrera. permite al usuario establecer metas concretas, así como un variado catálogo de recompensas e interacciones sociales



Figura 4. Aplicación móvil Nike Running

² <https://habituca.com/static/home>

³ https://play.google.com/store/apps/details?id=com.nike.plusgps&hl=es_419&gl=US

- Makeblock mBot Neo⁴: Se trata de un robot diseñado para que los niños tengan un contacto inicial con distintos campos tecnológicos como la robótica, la electrónica y la programación. Es fácil de ensamblar y soporta lenguajes de programación sencillos como Python y Scratch. Junto al soporte de la plataforma Makeblock y sus piezas electrónicas desmontables y configurables basadas en Arduino permiten a los usuarios con más conocimientos crear robots mucho más complejos.

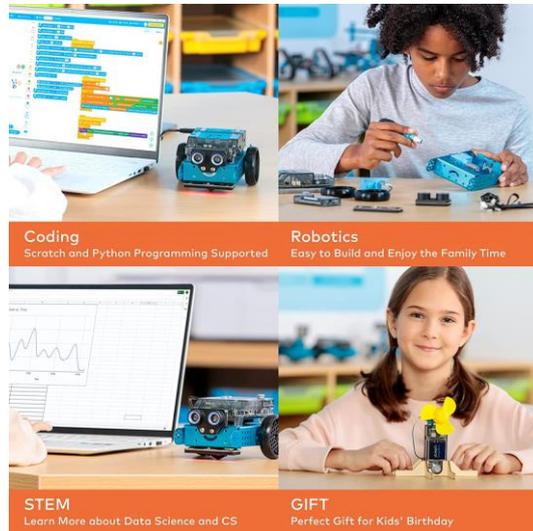


Figura 5. Makeblock mbot Neo

2.2 Aprendizaje Basado en Juegos

El aprendizaje basado en juegos se inspira en una metodología de enseñanza que utiliza el juego como herramienta didáctica y lúdica. Centrándose en el alumno y en su participación en el proceso de aprendizaje, fomentando la creatividad, el pensamiento crítico y la colaboración.

Cualquier juego puede ser utilizado como una herramienta didáctica orientada a las necesidades de enseñanza y los objetivos educativos de los docentes, despertando el interés y la motivación de los estudiantes para facilitar el aprendizaje de forma lúdica (Cornellà, Meritxell y David 2020).

(Shute y Ke 2012, p. 46) deduce siete elementos básicos que un juego bien diseñado debe poseer:

- Resoluciones interactivas: El juego debe plantear una serie de problemas o misiones con las que el jugador deberá interactuar y resolver.
- Es necesario que el juego plantee una serie de objetivos y reglas específicas que permitan al jugador centrarse en una meta.
- Retos adaptativos: Los buenos juegos equilibran los niveles de dificultad para adaptarse a las capacidades de cada jugador.
- Control: El jugador debe poder mantener cierto grado de influencia sobre el juego, su entorno y la experiencia de aprendizaje.
- Retroalimentación continua: Los buenos juegos deben proporcionar información oportuna a los jugadores sobre su rendimiento, para ayudar y estimular el aprendizaje.

⁴ <https://store.makeblock.com/products/diy-coding-robot-kits-mbot-neo>

- La incertidumbre: provoca compromiso y curiosidad por parte del jugador. Si un juego se considera predecible, perderá su atractivo e interés.
- Los estímulos sensoriales: se utilizan como una herramienta para excitar los sentidos, utilizando una combinación de elementos gráficos y sonidos para captar la atención del jugador.

En la actualidad existen una multitud de juegos educativos, tanto en papel como en formato digital, que pueden ser utilizados en el aula de forma fácil y rápida por parte de los docentes para estimular el proceso de aprendizaje, algunos de ellos son:

- Dragonbox⁵: una serie de juegos digitales centrados en la enseñanza de las matemáticas en los que los estudiantes deben resolver ecuaciones para avanzar, fomentando habilidades como la resolución de problemas, el razonamiento lógico y la manipulación de cifras y símbolos



Figura 6.Dragonbox

- Cities Skylines: Es un juego donde el jugador debe construir carreteras, zonificar áreas y proporcionar servicios a los ciudadanos. El juego plantea diversos temas como la planificación urbana sostenible, las dificultades de la creación, la gestión y el mantenimiento de una ciudad.



Figura 7.Cities Skylines

⁵ <https://dragonbox.com/>

- Minecraft Education Edition: Es una versión educativa del popular juego de construcción Minecraft. Está diseñado para enseñar a los niños una serie de temas, como ciencias, matemáticas, química, física, historia y lengua. Se puede jugar en modo individual o multijugador, y hay una versión especial del juego diseñada para las aulas.



Figura 8. Minecraft Education Edition

- Civilization 6: Es un increíble videojuego educativo, ofrece una gran cantidad de información sobre la forma en que funcionaban las civilizaciones antiguas. Al jugar, los jugadores pueden aprender mucho sobre historia, geografía, economía y diplomacia.



Figura 9. Civilization IV

2.3 Impacto de los videojuegos

Los videojuegos están basados en la interacción entre una o varias personas y un aparato electrónico que ejecuta dicho juego.

Según una infografía de DFC Intelligence, alrededor de 3.100 millones de personas juegan videojuegos como una forma de entretenimiento de forma habitual, esto representa alrededor del 40% de la población mundial («Global Gamers by Region - DFC Intelligence» 2021).

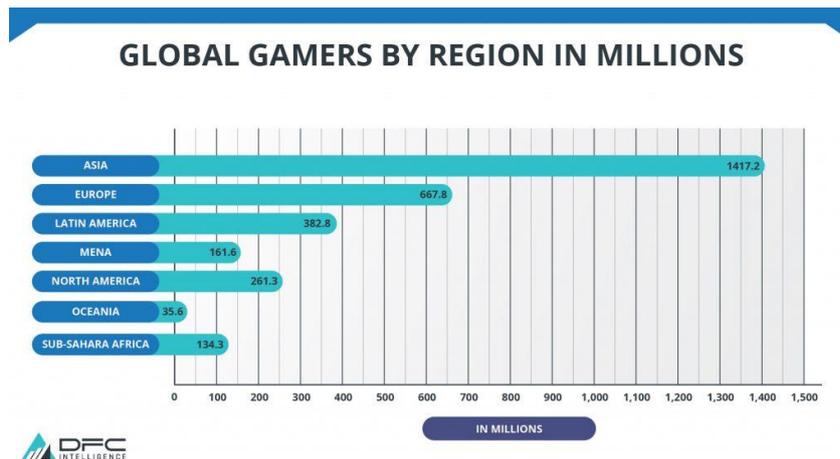


Figura 10. Número de jugadores por región en el mundo

Conforme a un estudio de la Asociación Española de videojuegos (AEVI), Se han convertido en una de las opciones de ocio audiovisual y cultural predilectas en España con una media de tiempo dedicada a los videojuegos de 8,1 horas semanales.

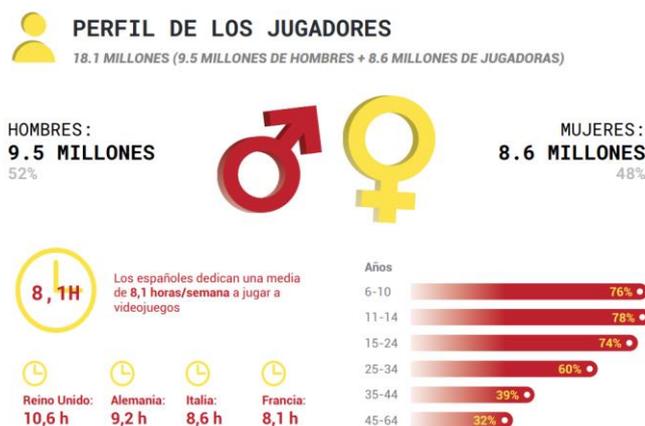


Figura 11. Tiempo dedicado a videojuegos (Asociación Española de Videojuegos 2021)

Según Daniel Parente.

[...] Esta experiencia de uso y la capacidad de conducir a los usuarios de forma rápida, concisa y eficiente, además de divertida a través de una serie de acciones predefinidas, es precisamente lo que mejor hacen los videojuegos y, por eso, son capaces de contar con millones de jugadores, con alto grado de diversión, que a diario dedican varias horas a los videojuegos (Ruth S. Contreras Espinosa y Jose Luis Eguia (Eds.) 2016, p. 11).

Por este motivo, el videojuego Danger puede llegarse a convertir en una herramienta didáctica para ofrecer una experiencia más inmersiva y envolvente que una otros métodos de enseñanza como una charla sobre mediadas para la prevención de incendios, ayudando a mantener la atención de los jugadores durante más tiempo, además gracias a internet, es posible propagar este videojuego como una herramienta didáctica por todo el mundo.

3. Herramientas

3.1 Introducción

Para poder llevar a cabo un proyecto grande es necesario utilizar muchas herramientas, tanto *software* como *hardware*: un motor de videojuego, un entorno de desarrollo integrado, un controlador de versión y un *framework* para el desarrollo de la página web. Por este motivo se dará una breve introducción de las distintas herramientas empleadas en el proyecto.

3.2 Motor de Videojuego

Un motor de videojuego proporciona una serie de herramientas de programación que permiten la creación, el diseño y el desarrollo de un videojuego. Existe una cantidad bastante diversa de motores de videojuego disponibles en el mercado, a continuación, se realizará un estudio de los motores más populares para el desarrollo de videojuegos 3D.

Según un estudio efectuado por Statista (Program Ace 2021), los motores de videojuegos independientes ocupan una gran porcentaje del mercado por parte de los desarrolladores de videojuegos 3D a la hora de crear un videojuego, pero existen muchos más que podrían ser utilizados a la hora de desarrollar el videojuego 3D de simulación de incendios.

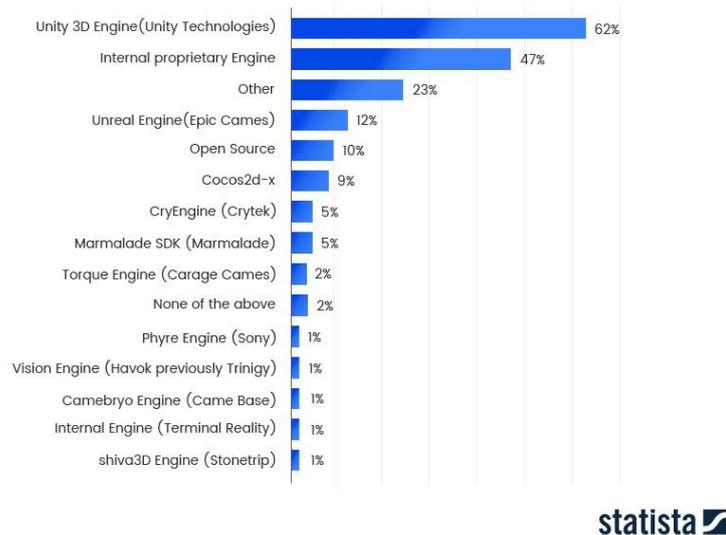


Figura 12. Gráfico de porcentaje de videojuegos creados con distintos motores

A la hora de decidir en qué motor de videojuego es necesario hacer un análisis del tipo de juego que se va a diseñar. Por ejemplo, si se quiere crear un juego de carreras, se necesitará un motor de juego que tenga la capacidad de manejar gráficos de alta calidad y efectos especiales. En cambio, un juego de estrategia em tiempo real como Danger no precisará de un motor de físicas muy potente pero será necesario que tenga la capacidad de calcular variables de manera eficiente, ya que a menudo manejan y calculan una gran cantidad de variables y estadísticas de forma simultanea

Dada la gran variedad de motores de videojuego que se encuentran en el mercado, es importante seleccionar aquel que cubra los principales requisitos que presenta el videojuego Danger.

- La capacidad de manejar una gran cantidad de partículas es necesaria para crear una simulación realista del fuego.
- Debe tener un motor de simulación de físicas 3D y es preferible que tenga una documentación clara sobre los métodos que implementa, puesto que se usará para implementar detecciones y colisiones con otros objetos.
- Debe tener soporte para establecer conexiones a internet. Se necesitará poder enviar y analizar las estadísticas del videojuego enviándolas a la página web.
- Debe ser multiplataformas, puesto que uno de los objetivos finales del proyecto es adaptarlo a la ejecución sobre dispositivos móviles.
- Es recomendado trabajar en algún motor que tenga una amplia comunidad, con una gran cantidad de assets y una documentación clara, puesto que ayudará a mitigar dos problemas que se pueden presentar en el desarrollo del videojuego: La falta de familiaridad con el motor de videojuego y la falta de experiencia en la creación y el modelado de objetos 3D.

Los tres motores más populares para el desarrollo de videojuegos 3D son Unity, Unreal y Cryengine.

Unity Engine

Los juegos creados con Unity⁶ han sido descargados más de 3,5 mil millones de veces desde su tienda de aplicaciones, y el motor gráfico de Unity está siendo utilizado por más de 5 millones de desarrolladores en todo el mundo. Algunos de los juegos creados en este motor son Hollow Knight⁷, Fall Guys⁸ y Pathfinder: Wrath of the Righteous⁹.

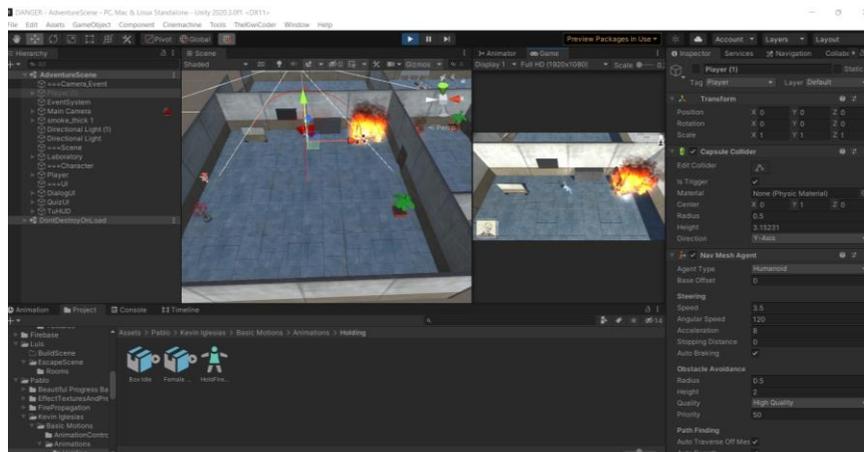


Figura 13. Interfaz de usuario de Unity Engine

⁶ <https://unity.com/es>

⁷ https://es.wikipedia.org/wiki/Hollow_Knight

⁸ <https://www.fallguys.com/es-ES/download>

⁹ https://store.steampowered.com/app/1184370/Pathfinder_Wrath_of_the_Righteous/

Unreal Engine

Unreal Engine es un motor de juego creado por la compañía Epic Games, se desarrolló principalmente para juegos shooters en primera persona, pero se ha utilizado con éxito en una variedad de otros géneros, incluyendo videojuegos de sigilo, lucha, MMORPG y otros RPG. Entre los juegos desarrollados se encuentran Fornite, BioShock y Borderlands.

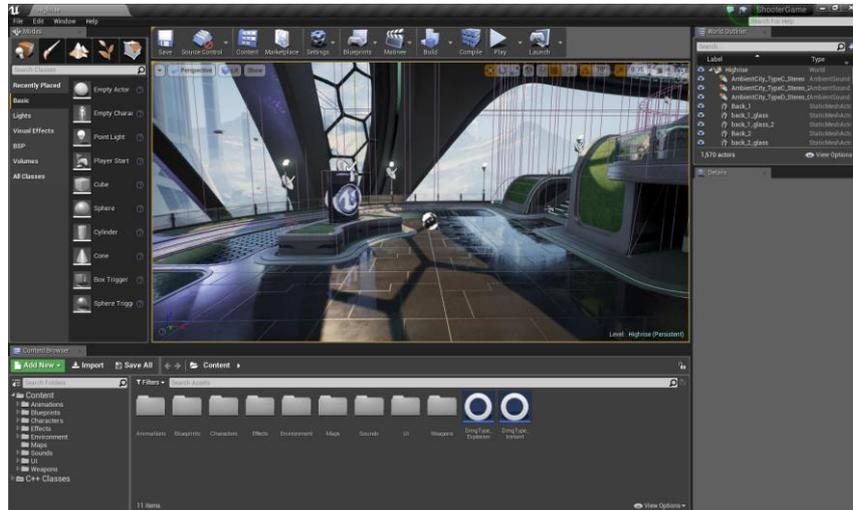


Figura 14. Interfaz de usuario de Unreal Engine

Cryengine

CryEngine es un motor creado por Crytek, creado con el objetivo de ser un motor muy adaptable, fácil de aprender y de usar. Tiene una de las características más potentes en el mercado de los motores gráficos, es capaz de generar un mundo en 3D totalmente en tiempo real. Se utilizó por primera vez en el videojuego Far Cry, entre otros videojuegos se encuentran Star Citizen, Hunt: Showdown, Kingdom Come: Deliverance.

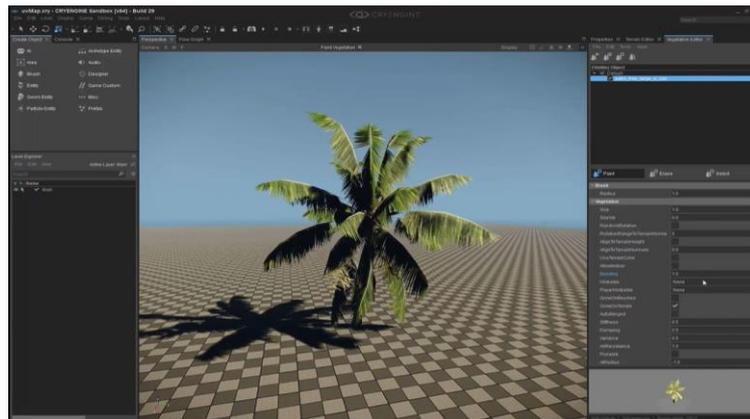


Figura 15. Interfaz de usuario de Cryengine

Conclusión

Desde el punto de vista de las capacidades de diseño, Unity es más fácil de usar que Unreal y Cryengine. Unity tiene una interfaz de usuario más intuitiva y una curva de aprendizaje más suave, haciendo de Unity uno de los motores más populares para desarrolladores independientes y estudios pequeños. También es más fácil de personalizar y tiene una mayor selección de herramientas y componentes.

En cuanto a las capacidades de rendimiento, Unreal y Cryengine son superiores a Unity. Ambas plataformas ofrecen un rendimiento mejorado y más consistente, especialmente en entornos 3D más complejos. Cryengine también ofrece una mayor flexibilidad en cuanto a la creación de contenido, lo que permite un mayor grado de control sobre el aspecto final de los juegos.

En cuanto a la documentación sobre el motor de videojuego, tanto Unity como Unreal presentan una documentación extensa y detallada que se encuentra disponible en sus respectivas páginas web. En el caso de Cryengine, aunque su documentación también se encuentra disponible en internet, esta es escasa en comparación con la de los otros motores de videojuegos lo que puede llevar a ralentizar el proceso de creación.

Hay que tener en cuenta que Danger se ha desarrollado con la herramienta Unity y el lenguaje de programación C#, por lo que el proceso de migración de un motor a otro puede ser costoso y en algunos casos contraproducente debido a las dificultades técnicas que presenta el proceso de migración.

Unity además cuenta con un gran número de recursos en línea para ayudar a los usuarios a familiarizarse y aprender a desarrollar con el motor. También hay un fuerte enfoque en el desarrollo móvil dentro de la comunidad y dado que se plantea añadir soporte para dispositivos móviles, esto supone una gran ventaja en comparación con los otros motores.

Debido a que el motor de Unity cumple con todos los requisitos necesarios que se presentan para el desarrollo del proyecto y dado el alto coste de un proceso de migración, se ha decidido seguir utilizando Unity como el motor de desarrollo para el proyecto.

3.3 Visual Studio (Entorno de desarrollo integrado)

Visual Studio es un entorno de desarrollo integrado (IDE), su propósito es el de simplificar la creación de aplicaciones y herramientas de software, así como la administración de la configuración y el código fuente.

Visual Studio ofrece una serie de ventajas para el desarrollo en Unity. En primer lugar, el IDE es muy intuitivo y fácil de usar, lo que permite a los desarrolladores concentrarse en el desarrollo de su juego en lugar de en la configuración del entorno de desarrollo. Además, Visual Studio ofrece una gran cantidad de herramientas y características para el desarrollo de juegos, como la capacidad de depurar y profundizar en el código, la opción de crear perfiles de rendimiento, integración de extensiones y el soporte para Unity Test Tools dentro del propio IDE.

Su principal característica es el uso de la arquitectura Modelo-Vista-Controlador (MVC), lo que permite a los desarrolladores crear aplicaciones web más eficientes y escalables.

Angular se caracteriza por su simplicidad, permite crear aplicaciones web utilizando una sintaxis sencilla y una estructura de directorios limpia. También es muy fácil de extender, puesto que cuenta con una gran comunidad de desarrolladores y una amplia gama de documentación y herramientas de apoyo, lo que facilita el desarrollo de aplicaciones web.

Debido a que la página web estaba siendo desarrollada en este framework y al tener algo de experiencia con TypeScript y JavaScript, se ha decidido continuar utilizando este framework para el desarrollo de la página web.

3.6 Firebase (Base de datos)

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles, integrada con Google Cloud Platform. Ofrece una base de datos en tiempo real, alojamiento de archivos, autenticación de usuarios y muchas otras características útiles para desarrolladores. Firebase es una plataforma muy versátil que se puede usar para construir una variedad de aplicaciones, desde juegos en línea hasta aplicaciones de redes sociales. Las principales razones por las que se ha seleccionado esta herramienta

- Permite sincronizar los datos de distintos proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
- Utiliza un conjunto de herramientas multiplataforma que permiten integrar fácilmente para plataformas web como en aplicaciones móviles. Es compatible con grandes plataformas, como IOS, Android, aplicaciones web, Unity y C++.
- Emplea la infraestructura de Google por lo que permite escalarla para adaptarse a cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.

Firebase ofrece una consola de administración centralizada, una base de datos NoSQL en tiempo real Cloud FireStore¹⁰, almacenamiento en la nube Cloud Storage¹¹, autenticación de usuarios Firebase Authentication¹² y muchas otras herramientas.

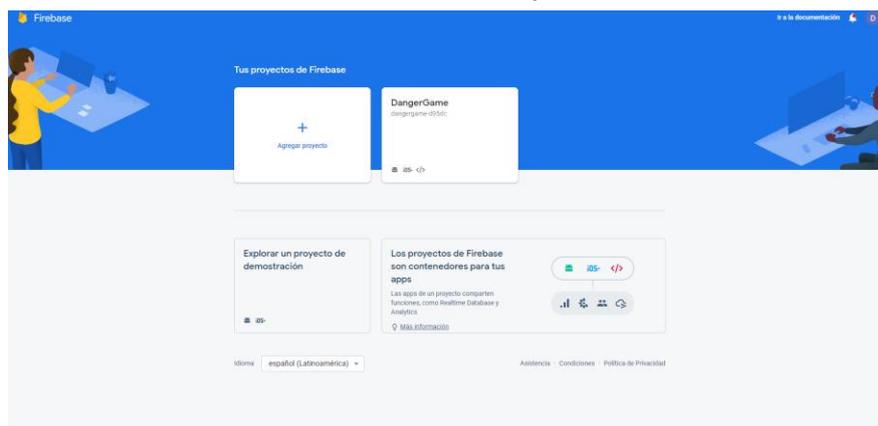


Figura 18. Interfaz de la consola de Firebase

¹⁰ <https://firebase.google.com/docs/firestore>

¹¹ <https://firebase.google.com/docs/storage>

¹² <https://firebase.google.com/docs/auth>



4. Análisis

4.1 Introducción

Tal como presentaba en la introducción, Adrián Sánchez Lavarias¹³ y Pablo Querol Ballester¹⁴ han diseñado y creado la estructura básica del juego, la cual ha sido ampliada posteriormente por Xinyu Jiang¹⁵ con el diseño y la implementación de un nuevo modo de juego.

El videojuego empieza con una cinemática introductoria y el menú principal, este menú contiene seis botones que corresponde a las funcionalidades de: créditos, configuraciones del juego, modo Construcción, modo Escape, modo Aventura y salir del juego.

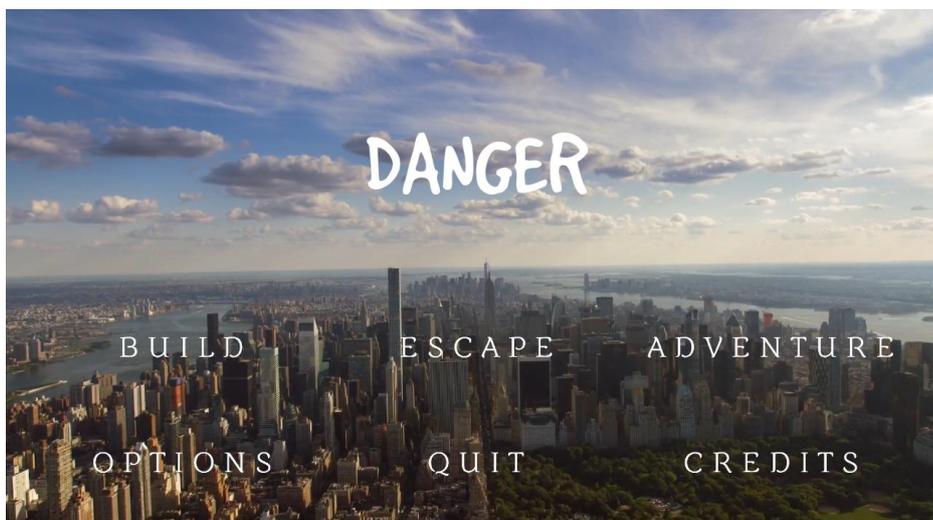


Figura 19. Menú de inicio del videojuego DANGER.

Los tres modos de juego conforman actualmente el núcleo didáctico y de aprendizaje del videojuego. Cada uno intenta plantear al jugador distintos retos y desafíos con el objetivo de que el jugador conozca las medidas de seguridad que se han de tomar, así como los procedimientos de emergencia en caso de incendios.

A continuación, se dará un breve repaso del estado de cada modo de juego, así como de explicar el objetivo educativo que intenta cubrir cada uno.

4.1.1 Modo Construcción

Este modo de juego diseñado por Adrián Sánchez Lavaria ha sido implementado con la visión de crear una habitación o una sala que cumpla con los requerimientos establecidos por la normativa de prevención de incendios.

El jugador dispone de un temporizador que permite saber el tiempo que lleva construyendo la habitación y un botón Scan que permite comprobar si la habitación que ha generado se adecúa a las normas de incendios, enviando la estadística del tiempo usuario ha empleado en construir la habitación a la página web mediante Firebase.

¹³ <https://riunet.upv.es/handle/10251/174398>

¹⁴ <https://riunet.upv.es/handle/10251/174568>

¹⁵ <https://riunet.upv.es/handle/10251/182255>

Además de un botón Reduce Exteriors que permite reducir las paredes para ver de forma más clara la habitación. Para poder construir las habitaciones, el usuario dispone de un menú situado en la esquina inferior derecha con los distintos elementos disponibles para crear la habitación.

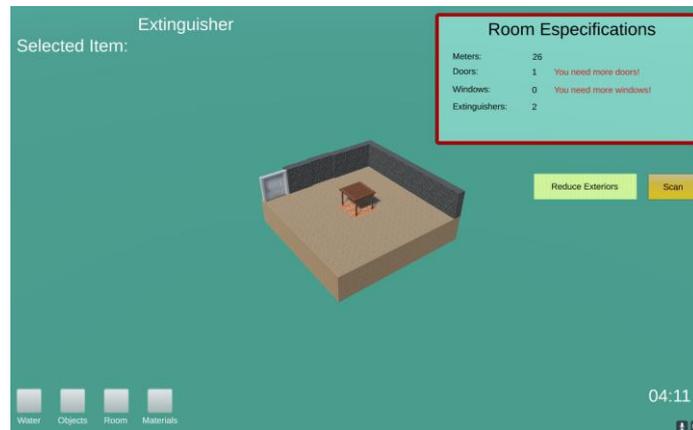


Figura 20. Modo Construcción

4.1.2 Modo Escape

El modo escape creado por Pablo Querol Ballester, se basa en un juego de estrategia en tiempo real (RTS) donde el jugador deberá ayudar a todos los personajes en la escena a escapar de un edificio en llamas. El nivel se genera de forma procedural, por lo que cada vez que el jugador vuelva a jugar el diseño del nivel será diferente. El jugador podrá controlar a cualquier personaje en la escena con el objetivo de salvar el mayor número de vidas, llevándolos a la salida antes de que el incendio se propague por todo el nivel.

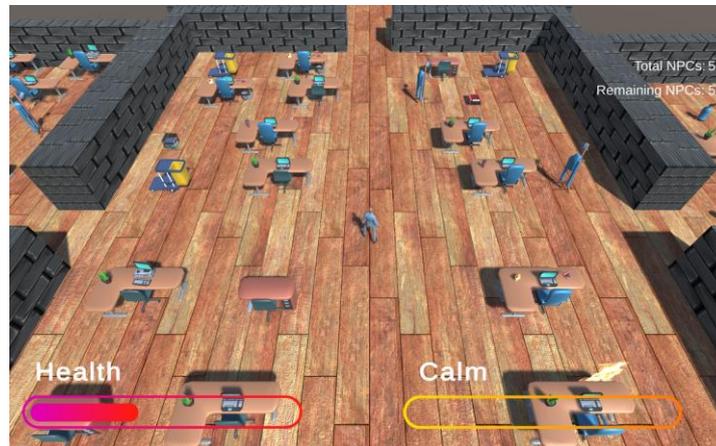


Figura 21. Modo Escape

4.1.3 Modo Aventura:

Diseñado por Xinyu Jiang, presenta un estilo de juego similar al Modo Escape, pero con una escena creada de forma manual, esto ha permitido crear un entorno donde es más fácil controlar el diseño de niveles y añadir elementos interactivos que ayudan a los jugadores a aprender.

Este modo de juego actualmente cuenta con nivel que cuenta con ocho habitaciones antes de llegar a la salida. En cada una de las habitaciones se presenta al jugador con breve cuestionario que tendrá que resolver de forma correcta para poder seguir avanzando, el objetivo es llegar a la salida antes de que el temporizador se acabe, respondiendo

correctamente preguntas relacionadas con acciones a realizar en caso de incendio y ayudando a los otros personajes a llegar a la salida con éxito.

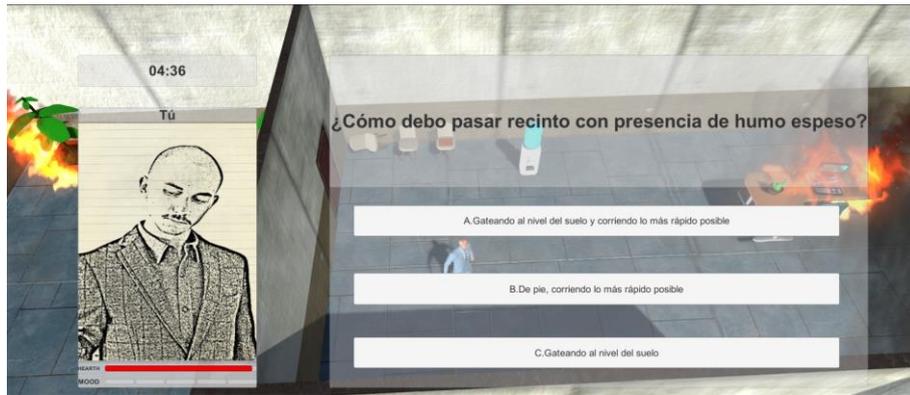


Figura 22. Modo Aventura

4.1.4 Página web

La página web se ha diseñado usando el *framework* de Angular, se compone de tres pestañas.

En la pestaña de información es posible encontrar documentación y manuales sobre las acciones a realizar en caso de emergencia, así como referencia a páginas web que proporcionan más información respecto a las medidas a tomar para la prevención de incendio, así como a información de las normas y medidas de prevención a seguir caso de incendio.

El apartado de historial mantiene una tabla con puntuaciones de usuarios en el modo Construcción. Implementado mediante Firebase, se establecen conexiones tanto con Unity como con la página web de Angular, almacenando la información de las puntuaciones generadas en el juego para mostrarlas en la página web.

En el apartado Test se encuentra una recopilación de todas las preguntas que se utilizan en el modo Aventura y sus respuestas.

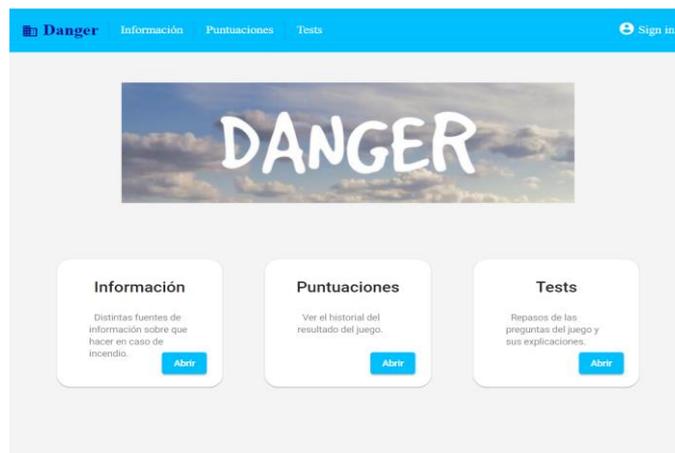


Figura 23. Portal de la Página web

4.2 Propuestas de mejoras

En este apartado, se procederá a analizar las distintas carencias que presenta el videojuego y los planteamientos que se proponen para mejorar los distintos modos de juego.

- El modo Construcción presenta muy pocos elementos con los que se pueden construir una habitación, además de no poder guardar el diseño de las habitaciones creadas por los jugadores.
- La generación procedural implementada en el modo Escape genera un escenario poco realista y con poca variedad de habitaciones, siendo muy similares entre ellas.
- El videojuego cuenta con un sistema de generación y propagación de fuego y humo, pero presenta algunas carencias, el fuego no está configurado para que pueda ser apagado ni pueda volver a propagarse.
- El modo aventura cuenta actualmente con un solo nivel de juego
- La tabla de puntuaciones implementada en la página web no proporciona mucha información útil sobre las estadísticas de cómo se han diseñado las habitaciones de los distintos jugadores.
- La página web no complementa al juego. La presencia de la página web es redundante, en la actualidad solamente ofrece información inútil para analizar y no complementa de ninguna manera el contenido del juego.

Para cada una de estas propuestas se procederá a listar los cambios que se procederán a realizar en cada modo de juego.

4.2.1 Modo Construcción:

Se pretende añadir nuevos **elementos de construcción** para permitir la creación de habitaciones más complejas. Además de editar la apariencia visual de varios elementos que se encuentran en el nivel, entre los que se incluyen ventanas, paredes, puertas y extintores. Para cumplir este objetivo se reutilizarán algunos de los assets los diferentes modos de juego junto a algunos así como algunos assets ¹⁶

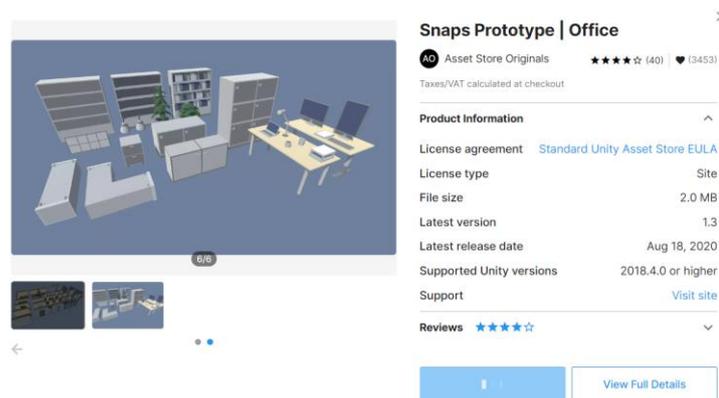


Figura 24. Assets de oficina

Se implementará un **sistema de guardado**, que permita a los jugadores crear y guardar las habitaciones que han diseñado. Además de crear las interfaces necesarias para cargar y guardar estos archivos a memoria.

¹⁶ <https://assetstore.unity.com/packages/3d/environments/snaps-prototype-office-137490>

4.2.2 Modo Escape: Mejorar Generación Procedural

Se desarrollará un nuevo **sistema de generación procedural** para las habitaciones debido a la poca flexibilidad que el sistema actual aporta para poder editar las habitaciones, haciendo que todas ellas sean muy parecidas entre sí. Este nuevo estará basado en la utilización de habitaciones prefabricadas, las cuales permitirán modificar y añadir variedad en cuanto al diseño de habitaciones para el generador de procedural. Además, se introducirán a las habitaciones nuevos elementos con los que el jugador podrá interactuar, como extintores y alarmas de incendio, con el objetivo de generar habitaciones más interactivas, realistas y cercanas a la experiencia real del usuario en la vida real.

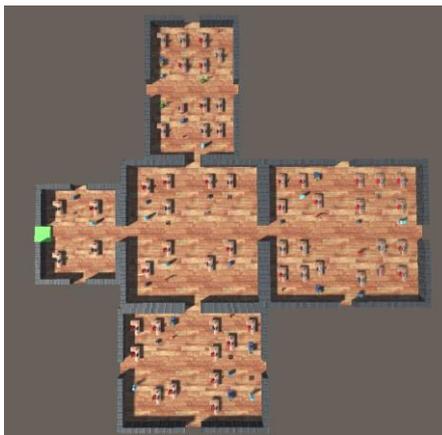


Figura 25. Nivel generado en el Modo Escape con poca variedad

También, **se modificará el sistema de generación de fuego y humo** para que sea dinámico e interactivo con el entorno los nuevos elementos a implementar en la escena. Como la utilización de extintores para poder apagar el fuego y que este se pueda volver a propagar o ralentizar la velocidad de propagación del fuego al activar alguna alarma contra incendios.

4.2.3 Modo Escape: Inteligencia Artificial

En el diseño utilizado para la Inteligencia Artificial, la detección del entorno se basa en el uso de Raycasts ¹⁷, líneas invisibles generadas con el objetivo de detectar colisiones con otros objetos.

Este método de detección permite obtener información sobre el objeto con el cual colisiona el rayo, pudiendo obtener información como el nombre, la etiqueta o algún componente o propiedad que contiene. Pero dificulta la percepción del entorno, ya que se ve fácilmente afectados por la cantidad y la dirección de los *Raycast*, pudiendo no detectar elementos que de otra forma los podría visualizar sin problemas. Por este motivo se **mejorará la detección del entorno por parte del personaje**. Utilizando diversas técnicas para asegurar que el personaje es capaz de detectar cualquier elemento dentro de su campo de percepción de manera correcta.

¹⁷ <https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>

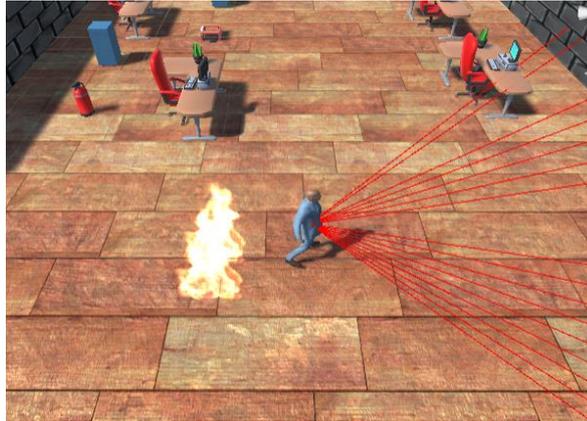


Figura 26. El NPC no es capaz de detectar el fuego que está en su espalda

En cuanto a la Inteligencia Artificial de los NPCs, **se implementarán y diseñarán árboles de comportamientos**, con el objetivo de expandir las acciones y comportamientos que pueden realizar los NPCs de una forma flexible y modular.

Un árbol de comportamiento es un modelo matemático de ejecución de planes, se basa en la utilización de un grafo dirigido de nodos, donde cada nodo representa una acción o una condición. El árbol se evalúa desde la raíz las hojas, el orden de ejecución de los nodos se realiza de izquierda a derecha

Normalmente los árboles de comportamiento están compuestos por dos tipos de nodos, los cuales pueden alternar entre tres estados distintos, éxito, fallo o en ejecución

- Nodos de control de flujo: Representan a los nodos internos de un árbol, se encargan de ejecutar las subtareas que contienen, así como de propagar el estado a otros nodos. Entre ellos los más importantes son:
 - Nodo selector: Ejecuta por completo al primer nodo hijo que no falle en su ejecución y en caso de fallar, pasarán al siguiente nodo hijo para intentar completar su ejecución.
 - Nodo de secuencia: los nodos de secuencia ejecutarán de forma secuencial a todos los nodos hijos que contenga, pero detendrán su ejecución en el caso de que uno de sus nodos hijos devuelva un fallo, propagando el estado de fallo hacia los nodos superiores
- Nodos de acciones: Representan los nodos hoja en un árbol y una acción muy básica que se pretende realizar, el conjunto de estas acciones organizadas por un nodo de control de flujo permite la creación de tareas complejas. Entre ellos también se encuentran los nodos que actúan como comprobadores de determinadas condiciones, propagando el estado de fallo a sus ramas superiores si no se llegan a cumplir o continuando su ejecución en caso contrario.

A partir de esta estructura se pueden elaborar una estructura que aporta flexibilidad y modularidad a la hora de crear nuevos comportamientos, pudiendo generar comportamientos complejos basados en una secuencia de acciones básicas realizadas de forma consecutiva. Una representación del comportamiento actual de los NPC puede ser representado por el siguiente árbol de comportamiento.



Figura 27. Árbol de comportamiento del personaje

Este árbol de comportamiento representa el posible comportamiento que podría implementar un NPC. Se comprueba si el personaje ha detectado fuego y procederá a alejarse del fuego hasta que este a una distancia moderada del fuego, a continuación, buscará la salida más cercana para salir del edificio. En caso de que no haya detectado fuego, el NPC deambulará por el mapa, moviéndose a un punto aleatorio, esperando una cantidad de tiempo tras el cual se volverá a reevaluar y comprobar la condición de detección del fuego.

4.2.4 Modo Aventura

Se diseñará el segundo nivel del modo aventura, contará con 9 habitaciones, en las que se presentará al jugador con cuestionarios o acciones que pueden realizar.

Este nuevo nivel y el ya existente implementarán las nuevas y mecánicas que irán implementando y añadiendo en los demás modos de juego.

Emergency Evacuation Plan

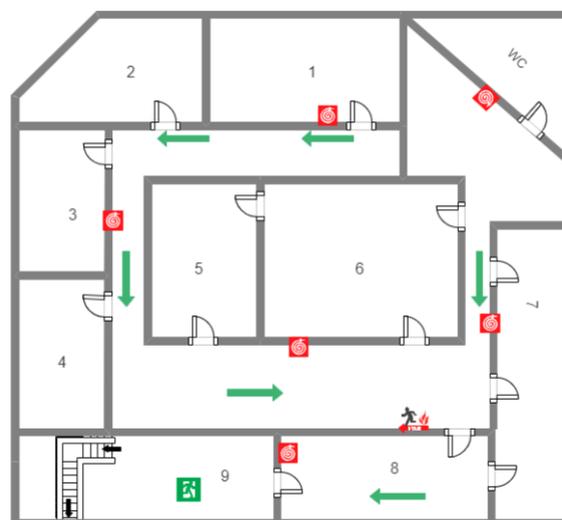


Figura 28. Esquema del nuevo nivel

La primera habitación será en lugar donde el jugador aparecerá, el jugador deberá apagar el fuego que se ha generado en la escena utilizando el extintor que se ha ubicado a la salida de la habitación. Poco después de que se apague el fuego empezará a crecer con más fuerza y el jugador tendrá que evacuar de la habitación.

En la segunda habitación del jugador deberá avisar a sus compañeros sobre el incendio, estos acompañarán al jugador hasta la salida. El jugador deberá acercarse a ellos para hablar.

En la tercera habitación el jugador tendrá que rescatar al NPC que ha sido rodeado por las llamas, teniendo que usar la carga restante del extintor, y en caso de no ser suficiente podrá tomar el que se encuentra en el pasillo de la habitación

En la cuarta habitación el jugador podrá encontrar un teléfono con el que podrá avisar a los bomberos para que acudan al edificio.

La quinta y sexta habitación contendrán más NPC con los que el jugador podrá interactuar para ayudarles a dirigirse a la salida.

Además, cerca de la quinta habitación se encontrará la alarma de incendios, que podrá ser activada.

La séptima habitación será una habitación en llamas, el jugador deberá estar atento a las señales que se proporcionan por el mapa para evitar entrar en ella y dirigirse la salida a través de la octava habitación.

En la octava habitación el jugador deberá pasar agachado por las cantidades de humo que se han generado desde la séptima habitación, en caso de que el jugador pase sin estar agachado la vida de jugador ira decayendo rápidamente.

En la novena habitación el jugador deberá apagar las llamas que se encuentran cerca de la salida, se ha colocado un extintor adicional en la puerta entre la octava y la novena habitación para que el jugador pueda completar con éxito el nivel.

Todo el camino hasta la salida estará señalizado con carteles que indicarán donde se encuentra la salida. El jugador podrá decidir dirigirse directamente a la salida, pero debido al sistema de puntuación recibirá menos nota si ayuda a todos los demás NPC a escapar.

Se implementarán **cambios relacionados a la cámara** utilizando Cinemachine¹⁸, un *asset* de Unity que permite utilizar cámaras dinámicas e inteligentes creando efectos de cámara complejo sin la utilización de gran cantidad de código. Página web: Catálogo de habitaciones

Aprovechando el diseño del sistema de guardado comentado en el apartado 4.2.1 , **se implementará un catálogo de habitaciones** donde los jugadores podrán subir desde el juego las habitaciones que hayan creado a la página web para que puedan ser visualizadas y descargadas por otros jugadores.

Se implementará un **sistema de autenticación y registro** utilizando Firebase, siendo como requisito que el jugador se encuentre registrado para poder subir sus habitaciones. El registro realizará desde el propio juego.

4.2.5 Página web: Cuestionario

Por otra parte se va a modificar el apartado Tests de la página web para que tenga la funcionalidad de un **cuestionario de preguntas y respuestas**

Las preguntas se almacenarán utilizando Cloud Firestore , una base de datos No SQL de Firebase, la página web de Angular podrá acceder a ellas, consultandolas y seleccionando de manera aleatoria diez de ellas para generar el cuestionario que se le mostrará al usuar

¹⁸ <https://unity.com/es/unity/features/editor/art-and-design/cinemachine>



5. Implementación

5.1 Introducción

En este apartado se explicará con detalle cómo se han realizado las implementaciones de las mejoras del apartado anterior. Se han introducido ciertos cambios en la estética del juego para adecuarla a una temática relacionada con los incendios.



Figura 29. Nuevo Menú principal

5.2 Modo Construcción

Se añadirán nuevos objetos para ampliar el modo construcción. Entre los cuales se encuentran: sillas, plantas, estanterías, papeleras y archivadores de oficina. Estos aportarán variedad al diseño de niveles del modo construcción.



Figura 30. Ejemplo de habitación creada

5.2.1 Mejoras en la interacción con el usuario

Dado que la forma de previsualizar un objeto realizaba mediante un cuadrado verde que indicaba la posición del objeto a colocar, no era sencillo previsualizar objeto y era difícil



Figura 31. Previsualización de una silla

5.2.2 Sistema de Guardado

Se aprovechará la estructura en la que se organizan los objetos en la escena para implementar el sistema de guardado. El Gameobject *Room*, almacenará todos los objetos que el jugador ha utilizado para crear una habitación y a partir de este objeto se llevará a cabo la serialización de la habitación, Unity pueda guardarlo en un archivo en memoria.

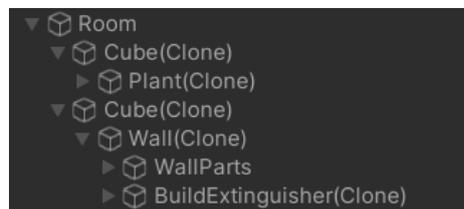


Figura 32 Estructura Del Gameobject Room

La organización jerárquica de Unity permite agrupar y ordenar los objetos en la escena. Room será el padre de todos los Gameobjects que formen parte de la habitación, los bloques que conforman el suelo de la habitación serán los hijos directos del objeto Room, dentro de cada uno de ellos se encontrarán los elementos que se pueden colocar encima del suelo en esa posición y en el caso de las paredes, estos podrán contener a su vez otros objetos como los extintores.

Unity solo permite la serialización de determinados tipos de datos ¹⁹, en este proyecto se utilizarán:

- Tipos de dato primitivos (int, float, string)
- Ciertos tipos de datos integrados en Unity (*Vector3*, *Quaternion*, etc.)
- Clases con la etiqueta *Serializable*
- Listas de datos y clases serializables

¹⁹ <https://docs.unity3d.com/es/current/Manual/script-Serialization.html>

Dado que no se puede serializar *Gameobjects* de forma directa, se tendrá que crear una nueva clase que implementa la etiqueta serializable llamada *SaveRoom* la cual contendrá todos los datos que se necesiten guardar en memoria:

- Listas de *Vector3*, *Quaternion* y *string* para guardar la posición, rotación y el tipo de suelo que contiene la habitación a guardar.
- La clase *FurnitureObjetct*, la cual contendrá listas con la posición, rotación y el tipo de objeto que sirve de decoración y en caso de ser una pared contendrá
- La clase *StatsRoom*, guarda las estadísticas de la habitación creada como los metros que ocupa, los extintores que contiene, etc.
- Imagen codificada en base-64 de la habitación, esta se usa como una pequeña previsualización que será útil para los jugadores.

Para guardar el archivo se utilizarán las corrutinas²⁰ de Unity, utilizadas para ejecutar funciones en segundo plano. Junto al archivo se tomará una captura de pantalla de la habitación, además de los datos a guardar, para almacenarlos en un formato JSON con la extensión “.dataRoom”. La ubicación por defecto en la que se almacenaran estos archivos será: “AppData\LocalLow\DANGER\DANGER”

Se ha diseñado diferentes interfaces para cargar y guardar las partidas. El proceso de carga es el más interesante debido a que se deberá obtener y mostrar las distintas partidas que se encuentran almacenadas en memoria. Pudiendo hacer clic en cualquiera de las imágenes para cargar la habitación que se ha seleccionado.



Figura 33. Interfaz de Guardado de Partida

El proceso de cargar las partidas de memoria se realizará en la clase *LoadSave*, utilizando una interfaz en la cual se cargarán aquellos archivos con la extensión “.dataRoom” almacenados en memoria, ordenándolos y paginándolos para cargar aquel que el usuario seleccione para su visualización.

²⁰ <https://docs.unity3d.com/es/530/Manual/Coroutines.html>

5.3 Modo Escape

5.3.1 Animaciones

Se modificarán los extintores para que generen partículas de humo, las cuales podrán detectar colisiones con elementos que representen el fuego. La colisión de ambas producirá que se dejen de generar nuevas partículas de fuego, causando que después de cierto tiempo el fuego se extinga.



Figura 34. Partículas del extintor activadas

La lógica relacionada con la colisión de las partículas de humo se realizará a partir de un script, utilizando las funciones de Unity que permite detectar colisiones entre partículas y otros Colliders, como el BoxCollider empleado en las partículas de fuego

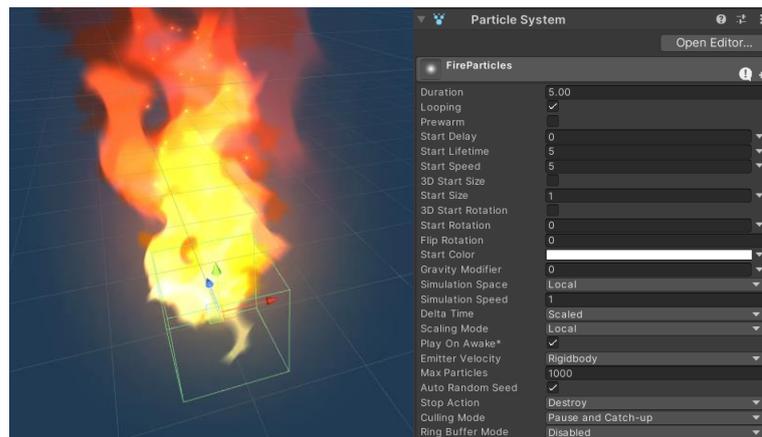


Figura 35. BoxCollider de la partícula de fuego (izquierda) y configuración de la partícula de fuego (derecha)

El uso de Colliders es de especial utilidad en el diseño de animaciones, utilizar colisiones permite modificar el comportamiento de determinadas animaciones a partir de fragmentos de código con el objetivo de cambiar el comportamiento de los objetos.

Se han animado las puertas para que se puedan abrir mediante una animación. La detección de colisiones mediante los Colliders permitirá alternar el estado en el que se encuentra la puerta, abriéndola cuando un personaje se encuentre cerca de ella y se detecten la colisión entre los dos objetos, y cerrándose cuando este se aleje y se dejen de detectar.



Figura 36. Puerta cerrada (izquierda) y Puerta abierta (derecha)

Se añadirán nuevas acciones que los personajes podrán realizar. Se modificarán las acciones de movimiento para que sean más fluidas a la hora de caminar implementando árboles de mezcla²¹.

Las animaciones se han conseguido de la página web Mixamo²² la cual proporciona un variado conjunto de ellas para importar a Unity de forma gratuita y con relativa facilidad. Se pueden subir el modelo de un personaje a Mixamo y descargar las animaciones para ese modelo, pero también es posible descargar nuevos personajes junto con animaciones desde la misma página web.

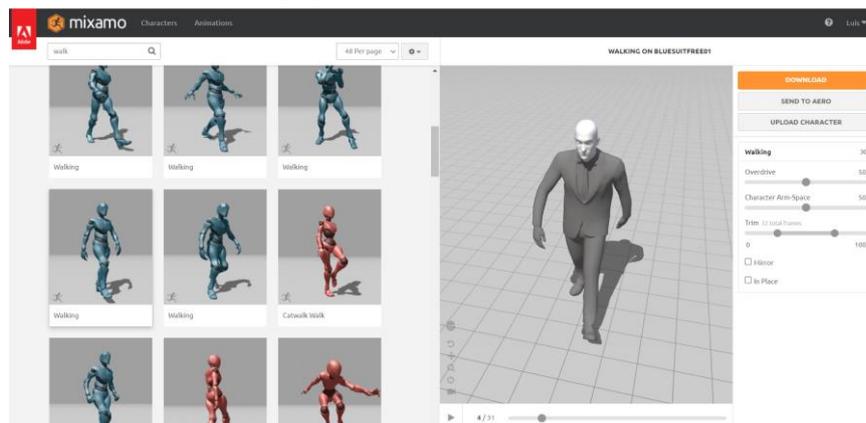


Figura 37. Personaje importado a Mixamo probando una animación de caminar

El componente Animator de Unity permite a los usuarios controlar la animación de un personaje o modelo 3D. Se puede aplicar a gameobject, y le permite al usuario crear y editar secuencias de animación para el objeto al que se encuentra vinculado. El componente se basa en una máquina de estados donde cada estado representa la animación que se reproducirá en cada momento y las transiciones pueden ser controlados por los distintos parámetros que se pueden definir. El componente permite configurar el proceso de transición entre las distintas animaciones, así como la posibilidad de aumentar la velocidad, o poner en bucle determinados estados.

²¹ <https://docs.unity3d.com/es/530/Manual/class-BlendTree.html>

²² <https://www.mixamo.com/#/>

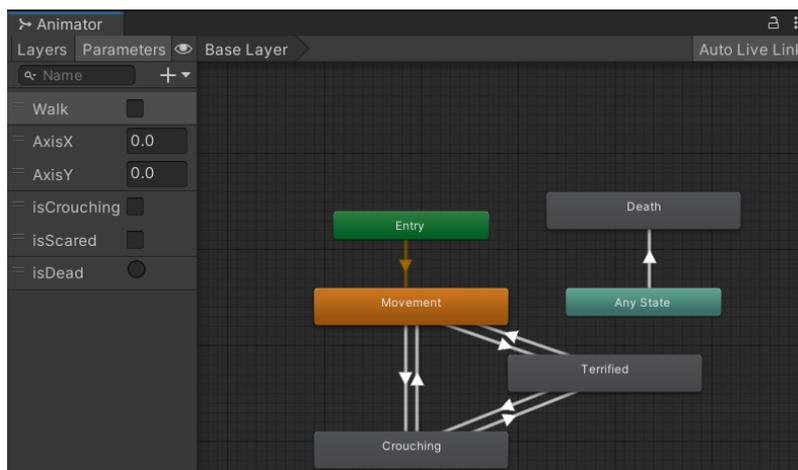


Figura 38. Componente animator de un personaje

Una de las técnicas más utilizadas en la animación de juegos es la mezcla entre dos o más animaciones. Unity proporciona los árboles de mezcla, los cuales permiten controlar las animaciones a partir de variables, lo que permite crear transiciones suaves y realistas, incorporando partes de todas ellas a diferente grado.

Para el control del movimiento del jugador se han implementado estos árboles de mezcla, utilizando dos variables donde “AxisX” controlará el movimiento lateral del personaje y “AxisY” el movimiento hacia delante o hacia atrás. Esta implementación permitirá alterar la animación de movimiento del personaje modificando estos dos parámetros a desde fragmentos de código referenciando a las variables.

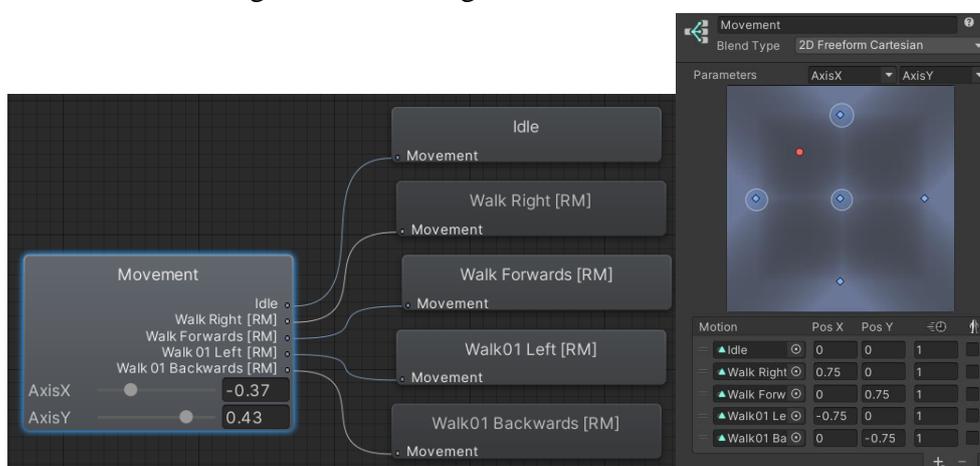


Figura 39. Árbol de Mezcla del estado Movimiento

Otra manera de conseguir mezclar distintas partes de una animación es el uso de una máscara de avatar²³. Por defecto cualquier animación utiliza una máscara completa la cual tiene todas las partes del cuerpo seleccionadas, y en consecuencia la animación se aplicará a todas las partes del cuerpo, pero es posible crear nuevas máscaras y modificarlas, esto permitirá que la animación vinculada a ella solo afecte a determinadas partes del cuerpo del avatar.

²³ <https://docs.unity3d.com/es/530/Manual/class-AvatarMask.html>



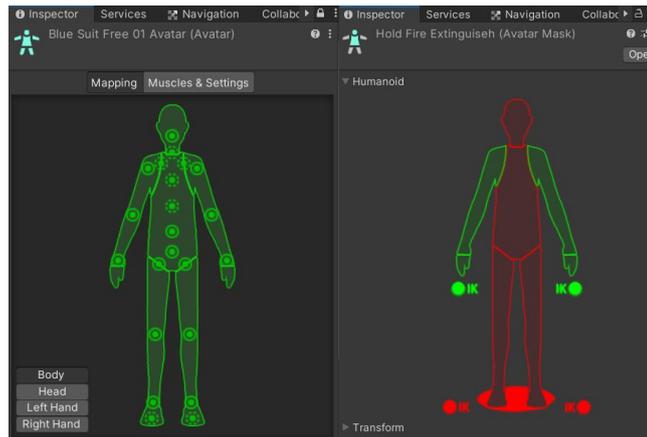


Figura 40. Avatar del personaje (izquierda) y Máscara del avatar para los brazos (derecha)

Para la implementación del uso del extintor se utilizará las Capas de Animación²⁴ que proporciona el componente Animator de Unity. Combinando la máscara que se ha generado en la Figura 40 junto con las Capas de Animación, se creará una animación que permita al jugador levantar los brazos del personaje simulando el uso de un extintor mientras que siga permitiendo al jugador moverse por el escenario. Esta nueva animación podrá ser configurada a partir de un script ajustando el parámetro Weight asociado a la Capa Extinguisher Layer.

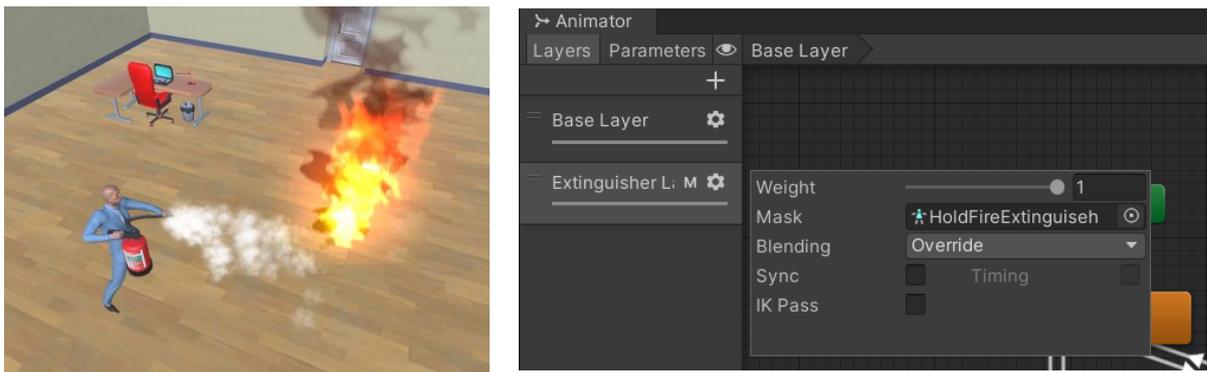


Figura 41. Animación de extintor(derecha) y configuración de la Capa Extintor en el componente Animator

Se añadirá una nueva animación que permitirá al jugador poder agacharse y moverse a nivel del suelo. Se implementará una mecánica que penalizará al jugador por moverse en zonas con una alta concentración de humo, haciendo que el jugador pierda rápidamente puntos de vida mientras se encuentre en estas zonas, agacharse ayudará al jugador a pasar por estas zonas de una forma más segura y sin recibir grandes cantidades de daño. La implementación de estas detecciones se explicará con mayor detalle en el apartado 5.3.2.

²⁴ <https://docs.unity3d.com/Manual/AnimationLayers.html>



Figura 42. Personaje en espera agachado(izquierda) y Personaje moviéndose(agachado)

La animación se añadirá al conjunto de movimientos de forma natural utilizando un árbol de mezcla similar al diseñado para el movimiento del personaje estando de pie. El jugador podrá cambiar de animación en cualquier momento mediante el uso de la tecla Ctrl, la cual será la tecla predefinida para realizar esta acción.

Por último, se añadirán dos nuevos estados que variarán dependiendo del estado del jugador:

- El estado de pánico: el jugador no podrá moverse ni interactuar con ningún elemento del escenario, sucediendo cuando el estrés del personaje llegue a su límite, reduciendo su nivel de estrés a un estado normal al terminar su efecto.
- El estado de muerte: esto sucederá cuando la vida de un personaje ha llegado a 0, el personaje entrará en coma y se desmayará, quedando inconsciente y muriendo poco después de que se termine esta animación.



Figura 43. Personaje Asustado(derecha) y personaje muerto (izquierda)

5.3.2 Sensores

Se creará un nuevo script que permita detectar con precisión todos los estados en los que podrá encontrarse el personaje con relación a su entorno. Este script tendrá como objetivo obtener la referencia de aquellos elementos que se encuentran alrededor del personaje y son relevantes para los cambios de comportamiento del personaje. Se utilizará los componentes de colisión de aquellos Gameobjects para devolver una lista filtrada con el objetivo de conocer su posición y su estado.

Una de las formas de detectar objetos es el uso de etiquetas para identificar a un conjunto de Gameobjects que pertenecen a una misma categoría, este es el método recomendado para detectar las colisiones entre objetos, en nuestro caso este método se usa para detectar si el personaje está cerca de algún fuego.

Muchos de los métodos proporcionados por Unity permiten pasarle como parámetro adicional aquellas determinadas Capas que sean relevante para realizar detecciones, en nuestro caso la función que se utiliza para crear un Raycast permite pasarle como parámetro adiciones cuales de las Capas serán utilizadas para detectar colisiones. Se puede cambiar la etiqueta y la Capa asociada al objeto desde el menú Inspector. Para usos prácticos se podrá configurar desde el script de sensor las Capas con las que podrán colisionar. Así como aquellas que serán opacas a efectos de la vista del jugador.

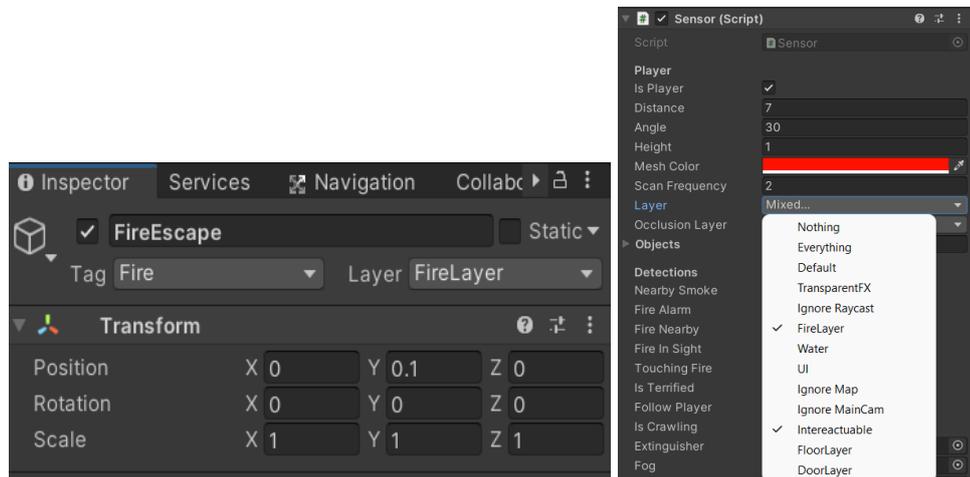


Figura 44. Etiqueta y capa del Gamobject fuego(izquierda) y Capas de detección del script Sensor

La configuración y el diseño de nuevos estados o detecciones que se pretenden observar se pueden configurar dentro del script “Sensor”. El modo de detección se dividirá principalmente en dos partes:

- La función scan: Encargada de realizar un primer escaneo de todos los objetos que se encuentran en el radio de detección del personaje. Almacenará en una lista la referencia a los distintos Gameobjects que se han detectado en cada escaneo. El uso de esta función puede ralentizar la ejecución del juego si no se usa con cuidado debido al alto coste computacional que requiera. Por este motivo el escaneo del entorno se realizará dos escaneos cada segundo, aunque es un parámetro que se puede configurar en el parámetro “ScanFrequency” dentro de las opciones configurables del script Sensor (Figura 44).
- La función isInSight: Permitirá pasarle como parámetro un Gameobject, utilizando la posición del personaje se lanzará un Raycast para poder averiguar si este Gamobject puede ser visto por el personaje o hay algún objeto que puede estar ocultando su visión.

Se creará una malla para representar el ángulo de visión del personaje, los objetos que podrá observar.

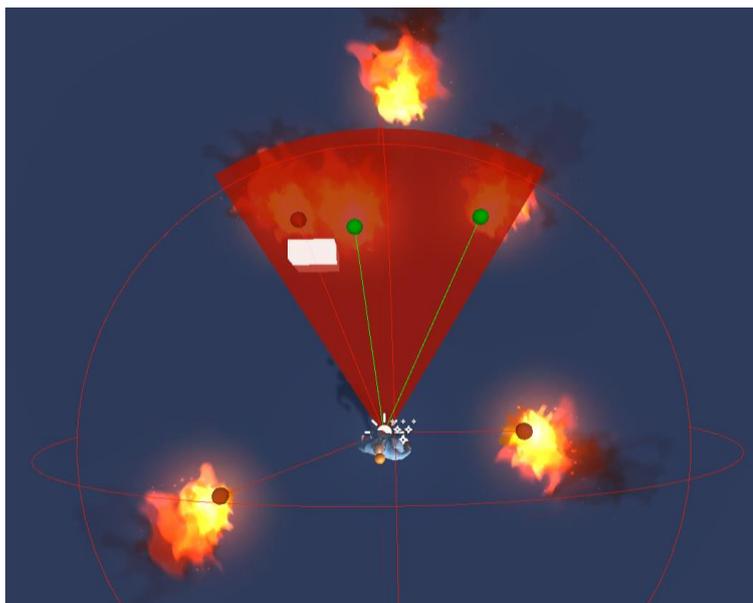


Figura 45. Ejemplo de detecciones implementadas por el sensor en caso de detectar fuego

El script “Sensor” mantendrá las detecciones que se encuentran alrededor del personaje, almacenando su estado en variables booleanas que podrán ser las cuales pueden ser consultadas desde otros fragmentos de código. Las detecciones que se observarán son:

- smokeNearby: Se activará si el personaje detecta una gran cantidad de objetos con la etiqueta Smoke, advirtiendo al jugador de que se encuentra en una zona con una gran cantidad de humo.
- fireAlarm: Se activará si se ha escuchado la activación de alguna alarma de incendios.
- fireInSight: Permite comprobar si en personaje puede observar fuego en su visión.
- nearbyFire: Utilizado para proporcionar una memoria a corto plazo sobre el evento de haber observado fuego durante un tiempo
- touchingFire: El jugador está pisando o tocando algún componente que contiene la etiqueta fuego por lo que estará recibiendo daño constantemente hasta que se aleje de estos componentes. Se utilizarán los componentes de colisión que lleva implementado el personaje.
- isTerrified: Se activará cuando el estrés del jugador supere el 100% de su capacidad inhabilitando el control del personaje durante los segundos hasta su recuperación
- isCrawling: Se activará cuando el personaje se encuentre gateando a nivel del suelo para reducir el daño que se recibe si se encuentra en una zona con una alta concentración de humo
- isDead: El personaje habrá muerto y no podrá ser controlado por el jugador, además de deshabilitar la inteligencia artificial asignada.
- followPlayer: Se activará como estado en el que algún NPC seguir al jugador a una cierta distancia.
- useExtinguisher: Se activará cuando un personaje está usando un extintor en la escena. También se controlará durante cuánto tiempo se ha utilizado el extintor para que cuando se agote su capacidad este deje de funcionar.

5.3.3 Inteligencia artificial

Para implementar los árboles de comportamiento se utilizará un asset creada por TheKiwiCoder²⁵, el cual permite la creación y visualización gráfica de un árbol de comportamiento en Unity, además de proporcionar plantillas para poder crear y configurar nuevos nodos en el árbol.

El modo escape está diseñado para que el jugador puede controlar a cualquiera de los personajes que se encuentren en la escena, por lo que es necesario que la inteligencia artificial implementada a partir de los árboles de comportamiento pueda activarse y pausarse en el momento en el que el jugador tome control del personaje.

El árbol de comportamiento se dividirá en varios comportamientos básicos, cada uno representará un conjunto de acciones que el personaje podrá llevar a cabo dependiendo del estado en el que se encuentre.

Cada rama del árbol representará una acción completa de que el personaje podrá realizar. Es posible añadir nuevas ramas, las cuales representarán nuevos comportamientos, así como modificar con facilidad la prioridad de cada acción y sus condiciones de activación.

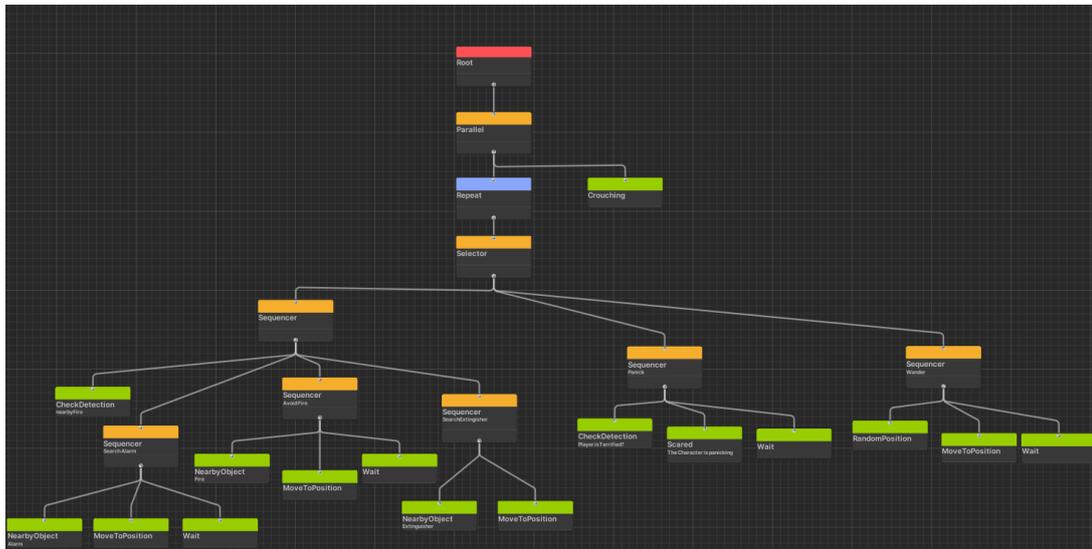


Figura 46.Árbol de comportamiento implementado para los NPC del modo Escape

Cuando un NPC pasa a modo autónomo, se activa el árbol de comportamiento que tiene asociado obteniendo el estado en el que se encuentra el personaje de la clase “Sensor”. Las acciones que podrá realizar el NPC configurado para la zona escape serán:

- El NPC intentará activar la alarma de incendio si ha observado fuego a su alrededor y no se ha activado esta alarma en el nivel, con el objetivo de reducir el ritmo de propagación del fuego.
- En caso de que la alarma de incendios se encuentre activada, el NPC intentará alejarse lo más posible del fuego, moviéndose a alguna habitación cercana que no esté en llamas si detecta fuego. En este caso no se dirige al NPC a la salida ya que la mecánica del nivel se basa en que el usuario guíe de forma segura a los NPC a las salidas señalizadas.

²⁵ <https://thekiwicoder.com/behaviour-tree-editor/>

- Si ha detectado fuego recientemente y el jugador se ha movilizadado a una zona segura. Podrá intentar buscar un extintor para utilizar.
- En caso de que un NPC este viendo un fuego y posea un extintor lo usará para intentar apagar el fuego.
- Si el estrés del NPC supera el límite que puede soportar entrará en un estado de colapso mental en el que no se podrá mover durante algunos segundos.
- Si no ha detectado fuego, el jugador se moverá a una posición aleatoria alrededor del personaje
- El caso de que la vida de un personaje llegue a cero, se ejecutará la animación de muerte, dejando al personaje tumbado en la escena sin que el jugador pueda interactuar con él.

Algunos de las funciones que se han empleado para los nodos con especial interés en cuanto a la lógica de utilización son:

Las detecciones sobre el entorno se obtendrán consultado a la clase “Sensor” y podrá ser invocadas desde el nodo “CheckDetection” pasando como parámetro el nombre de la variable a comprobar dentro de la clase “Sensor”.

Calcular la distancia a un objeto: Es necesario trazar el camino a recorrer sobre el Navmesh, sumando la distancia que este debe recorrer sorteando los obstáculos. En caso de que el camino hacia el objeto no se pueda completar por ser inaccesible, deberá ser descartado como un posible punto al que dirigirse.

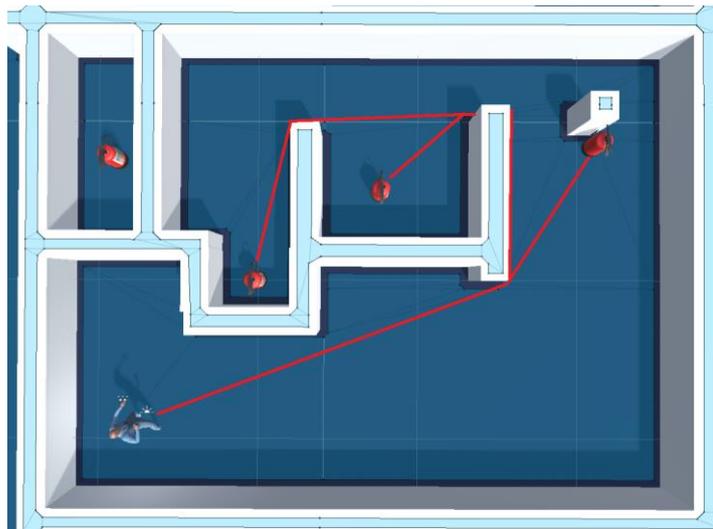


Figura 47. Cálculo de distancias usando el Navmesh generado por Unity

Para coordinar las acciones de movimiento, se ha creado un nodo denominado “MoveToPosition” el cual recibirá una posición a la que se deberán mover el personaje Unity. La implementación de este método necesita coordinar dos componentes de Unity:

- El componente Animator, la máquina de estados que nos permite modificar las animaciones de movimiento del personaje. Aunque este componente modifique las animaciones no desplazará al objeto por el escenario.
- El componente NavMeshAgent, permite al Gameobject vinculado moverse través el escenario usando el NavMesh.

En cuanto al método para encontrar un punto aleatorio al cual moverse, Se utilizará un método que permite obtener un punto aleatorio en un radio alrededor del personaje ²⁶ . Posteriormente se comprobará²⁷ que la nueva posición es alcanzable, obteniendo en caso contrario el punto más cercano se encuentra dentro del Navmesh.

5.3.4 Semilla Aleatoria

Para poder controlar la generación aleatoria de escenarios, se creará una Interfaz Gráfica que permitirá al jugador introducir los ajustes con los que se generará el mapa, entre como la semilla en forma de una palabra, con esta palabra los jugadores podrán repetir el escenario usando la palabra para volver a generar el mapa. La palabra se usará para generar un numero hash y se utilizará este número para inicializar la semilla del generador de números pseudoaleatorios.

En la interfaz se podrá seleccionar el número de habitaciones que se quieren generar en el escenario y el número de NPC que aparecerán en la escena y que el jugador deberá poner a salvo.



Figura 48. Interfaz de creación en el modo Escape

Se mantendrá el diseño de generación procedural basado en la construcción de las habitaciones sin utilizar prefabs y podrá ser activado desde la Interfaz Gráfica de generación del nivel activando la opción Legacy Mode.

5.3.5 Generación Procedural

El planteamiento se basa en la utilización de prefabs de las distintas habitaciones que se pretenden utilizar en la generación procedural del juego. A partir de estos prefabs, se pueda generar un mapa del juego de forma aleatoria compuestos por estas habitaciones. Actualmente la generación procedural contará con ocho habitaciones que se han diseñado para contar con número lo suficientemente variado para el generador procedural, pero es posible añadir, quitar o modificar las habitaciones que se han creado.

²⁶ <https://docs.unity3d.com/ScriptReference/Random-insideUnitSphere.html>

²⁷ <https://docs.unity3d.com/ScriptReference/AI.NavMesh.SamplePosition.html>



Figura 49. Prefab de una de las habitaciones

El prefab de una habitación deberá poseer dos tipos de scripts que se usarán para organizar y recuperar los datos de la habitación:

- El script Doorway, este script se deberá añadir a las puertas, se encargará de reemplazar la puerta por una pared y sellarla en cada de que la puerta no pueda conectarse con alguna otra habitación, además proporcionará ayuda visual trazando una pequeña línea roja para permitir detectar la orientación de la puerta.
- El Script RoomDetails, este script deberá estar presente objeto raíz del prefab, se encargará de mantener una lista de las puertas de una habitación y calcular el tamaño de una habitación.

El proceso para calcular el tamaño de una habitación se ha implementado mediante el uso de los métodos Bounds²⁸ que proporciona Unity. Este método permite obtener una variable con el tamaño que ocupa el objeto en la escena.

Para calcular el tamaño de la habitación es necesario que se detecten todos los GameObject hijos que la habitación contenga en su jerarquía e incluirlos dentro de la variable que guardará el área total. Se almacenará esta área en una estructura llamada “ocupiedArea” almacenando las coordenadas máximas y mínimas en los ejes X y Z de la habitación, estos puntos representarán el área que ocupa la habitación en la escena y ayudará a calcular la disponibilidad de las habitaciones en el nivel.

El proceso de creación de un nivel se realizará en los siguientes pasos:

1. Se generará la primera habitación, calculando el área que ocupa en la escena, almacenando en la lista “occupiedAreas” el área ya no se encuentra disponible para otras habitaciones.
2. Se seleccionará de manera aleatoria la siguiente habitación a colocar en la escena. Las habitaciones disponibles del generador se dividirán entre aquellas que tienen tres o más puertas y aquellas que tienen menos. Esto permitirá evitar casos donde no queden puertas disponibles por una generación con habitaciones que no contienen suficientes puertas para seguir generando nuevas habitaciones.

²⁸ <https://docs.unity3d.com/ScriptReference/Bounds.html>

3. Se recorrerá de manera aleatoria la lista de las puertas que todavía no han sido conectadas con otra habitación seleccionando una de ellas.
4. Se conectarán las dos habitaciones este proceso consiste en alinear (rotar y trasladar en el espacio 3D) las habitaciones, de manera que ambas puertas se encuentren en la misma posición, trasladando la nueva habitación generada a esta posición, posteriormente se rotará la habitación en el eje Y hasta que ambas puertas se encuentren en sentido opuesto.
5. Una vez alineadas se ejecutará las funciones para determinar si el tamaño que ocupa la habitación se encuentra disponible. En caso contrario se repetirá el proceso desde el paso tres. Si pese a esto no es posible instanciar la habitación se descartará la habitación
6. Cuando número de habitaciones que se pretendían crear se complete. Se hará un recuento del número de habitaciones creadas, si este número es inferior a un porcentaje configurable (actualmente un 60%) de las solicitadas, se eliminará el nivel y se repetirá el proceso de creación del nivel.
7. Se procederá a instanciar de forma aleatoria a los personajes en las distintas habitaciones de la escena, sustituir las puertas por las salidas y sustituir por paredes las puertas restantes.
8. Se activará el generador de fuego, el cual se encargará de propagar fuego por la escena y se dará comienzo al nivel.

Para la creación de las salidas del nivel se utilizarán cuatro de las puertas que no hayan sido utilizadas para conectar con otras habitaciones. El nivel tendrá una salida por cada una de las cuatro esquinas y se encontrarán lo más alejadas posible del centro del nivel.



Figura 50. Nuevo generado procedural

5.3.6 Generación de fuego

Se han implementado algunos cambios a la generación del fuego. El fuego se controla a partir de una Matriz de dos dimensiones. Cada casilla corresponde a un punto donde el fuego se puede propagar si la casilla no está en llamas.

Para conseguir una generación del fuego dinámica se modificará el script “FireGeneration”, este se encargará de mantener y controlar los lugares con fuegos

activos y la generación de las llamas iniciales, cada llama generada contará con un script “FireScript” el cual se encargará de propagar el fuego a las casillas adyacentes.

Las partículas de fuego pueden ser extinguidas usando los extintores que se encontrarán por las distintas habitaciones y en caso de que sean apagadas, notificarán a “FireGeneration” sobre la posición donde se encontraba el fuego para notificar que la casilla ha sido liberada del fuego. Si queda algún fuego a su alrededor, estos podrán volver a propagar el fuego a las casillas donde se ha apagado.

Las alarmas que se encuentren por el mapa podrán ser activadas para ralentizar la propagación del fuego, duplicando el tiempo que el fuego tarda en propagarse de una casilla a la siguiente.

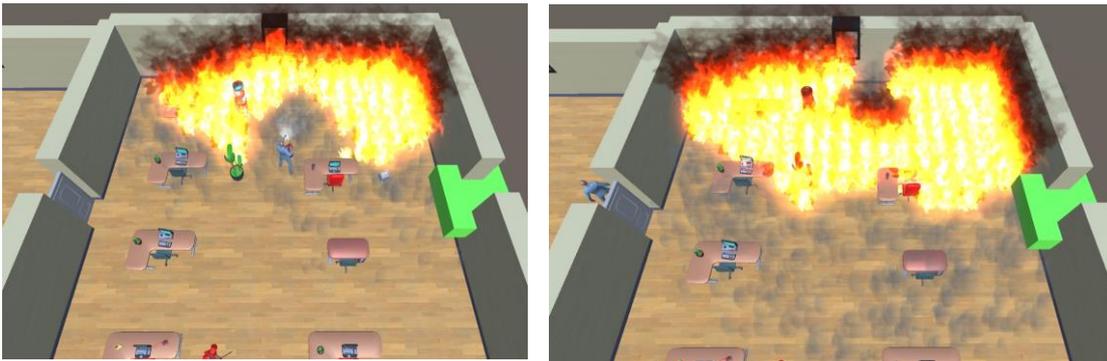


Figura 51. Personaje Apagando casillas del fuego(izquierda) y propagación del fuego unos segundos despues(derecha)

5.4 Modo Aventura

Se modificará algunas de las funciones implementadas dentro del nivel. Se implantarán tanto la utilización de sensores como las nuevas animaciones y acciones que los jugadores pueden realizar, así como la inteligencia artificial que se implementó para los NPCs en el modo Escape.



Figura 52. Detección de sensores en el modo Aventura

Se modificó la zona de humo y como interactuar como el jugador, ahora el humo causa mucho daño al jugador si el jugador no pasa agachado por debajo de él, como indica la pregunta que se le plantea al jugador antes de cruzar la barrera de humo.

En cuanto a los NPCs que había en la escena, se han modificados para que usen el diseño implementado de árbol de comportamiento

A diferencia del modo escape, en este modo de juego presentan un comportamiento mucho más sencillo y la funcionalidad principal que tienen será deambular mientras el jugador no interactúe con ellos, pudiendo seguir al jugador a la salida una vez el jugador interactúe con ellos y responda a las preguntas que se le plantean.

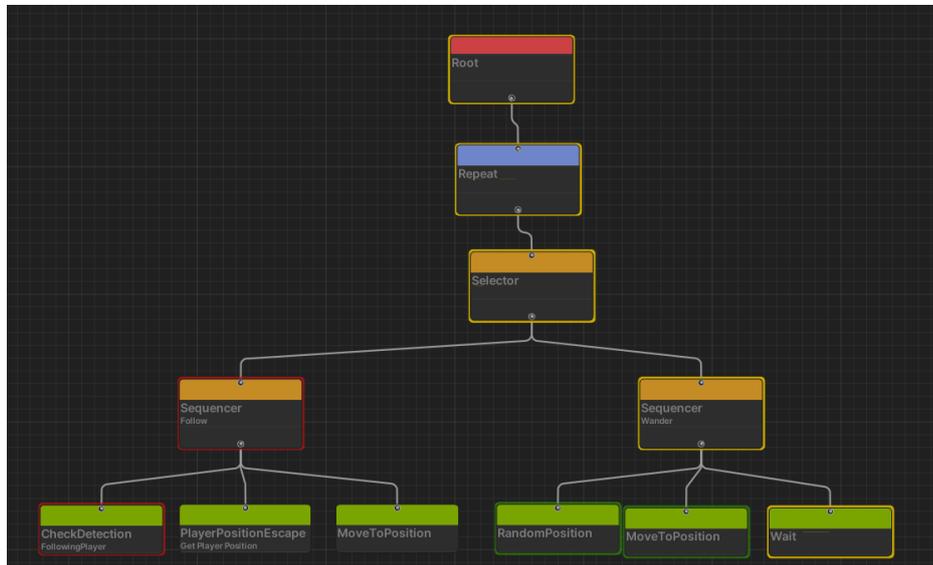


Figura 53. Árbol de comportamiento de los NPC en el modo Adventure

También se han modificado los extintores de la escena, implementando la funcionalidad que estos para que una vez activados puedan ser utilizados por el jugador para apagar el fuego, los extintores servirán para que el jugador pueda hacerse paso a través del pequeño fuego junto a la salida.

Haciendo uso de los assets de Unity Cinemachine se implementará el seguimiento del personaje, la rotación de la cámara y la funcionalidad de acercar y alejarla.

Por último, se implementará el nuevo nivel descrito en la zona de diseño. El nivel implementará el mismo sistema de puntuaciones, mecánicas y cuestionario de preguntas que contiene el primer nivel. El tiempo para completar el nivel será de cuatro minutos.



Figura 54. Nuevo nivel Modo aventura

5.5 Página web

Para poder lanzar la página web de forma local se debe tener instalado Nodejs el cual servirá de entorno de ejecución para JavaScript. Además de la extensión para ejecutar líneas de comandos de Angular desde una terminal (Angular CLI). Se puede comprobar la versión de NodeJs instalada en el ordenador utilizando el siguiente comando:

```
node -v
```

A continuación se llevará a cabo la instalación de Angular CLI a partir del sistema de gestión de paquetes de NodeJs (npm) usando el siguiente comando en una terminal:

```
npm install -g @angular/cli
```

Se necesitará ejecutar el siguiente comando en la ubicación donde se encuentran los archivos que conforman el proyecto de Angular:

```
npm install
```

Esto permitirá instalar todas las dependencias necesarias para poder alojar la página web de forma local, estas dependencias se en el archivo package.json .

La página web se puede lanzar con el siguiente comando:

```
ng serve -open
```

Este comando cargará los archivos necesarios para carga el programa y ejecutarlo en el navegador por defecto del ordenador, se puede acceder a <http://localhost:4200/> para visualizar la página web.

5.5.1 Cuestionario

Se cambiará el diseño implementado en el apartado de prueba, para que la interfaz sea la de un cuestionario de preguntas sobre prevención de incendios. El usuario deberá contestar correctamente la pregunta antes de pasar a la siguiente.

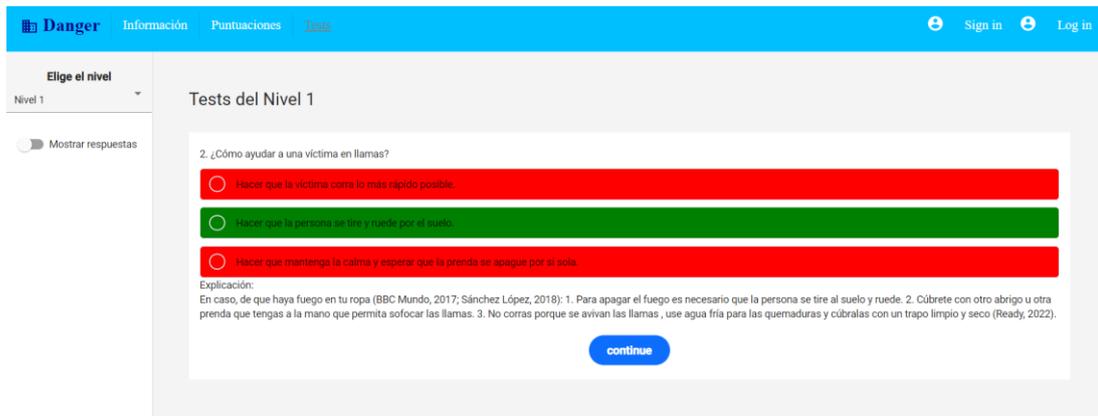


Figura 55.Nueva interfaz del cuestionario

Las preguntas se recuperarán de la base de datos Cloud Firestore, una base de datos no SQL que proporciona Firebase para almacenar información. Cada pregunta se almacenará con un identificador único que identificará a la pregunta.

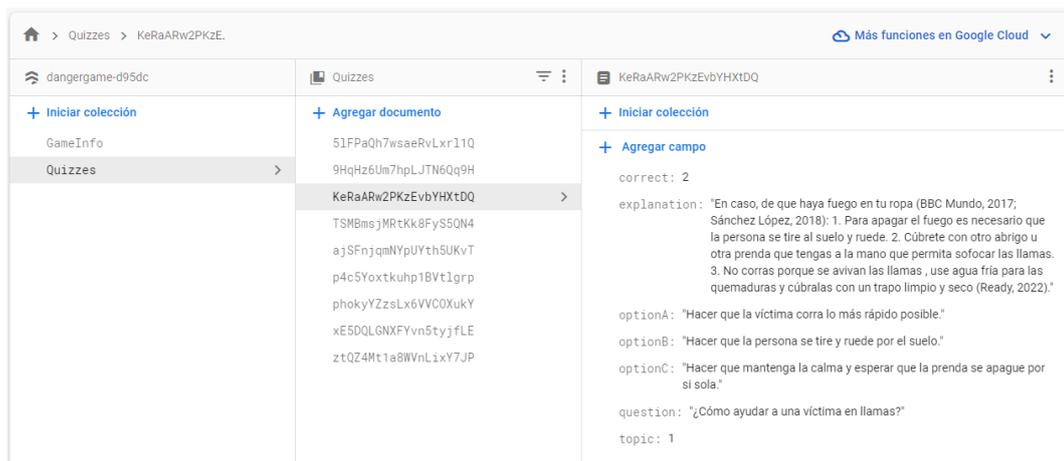


Figura 56. Interfaz de usuario

Cada documento dentro de la colección Quizzes representará una de las preguntas que podrá ser seleccionada para mostrar al jugador. Las preguntas se podrán categorizar por temas, aunque de momento solo existirá el tema relacionado con planes de actuación de emergencia, pudiéndose añadir nuevos temas posteriormente sobre preguntas relacionadas con incendios forestales o primeros auxilios en situaciones de incendios.

Para poder recuperar la información desde la base de datos habrá que realizar antes unos pasos previos. Se pueden encontrar los pasos a seguir en la documentación sobre como agregar el SDK de Firebase ²⁹ al proyecto de Angular.

Los pasos necesarios para poder llamar al servidor incluyen:

- Instalar los paquetes de Firebase usando NodeJS o cualquier otro gestor de paquetes
- Agregar la configuración personal del proyecto, esto incluye la configuración de la clave API privada del proyecto Firebase.

Para establecer la conexión con Angular, se han diseñado distintos servicios que permitirán recuperar la lista de preguntas desde la Cloud Firestore. La forma de recuperar información de Firebase se realizará mediante el uso de promesas.

Las promesas se usan para manejar valores asíncronos, y son comunes en código que hace llamadas a API externas. Representan el resultado eventual de una operación asíncrona.

5.5.2 Catálogo de habitaciones

Para poder recuperar la información desde la base de datos habrá que realizar antes unos pasos previos para poder agregar la SDK de Firebase a Unity³⁰. Este proceso es sencillo y solo se necesita:

- Agregar el archivos de configuración google-services.json al proyecto de Unity
- Instalar los paquetes SDK con los productos de Firebase a utilizar. Dentro de unity se utilizarán FirebaseAuth , Firestore y FirebaseStorage.

²⁹ <https://firebase.google.com/docs/web/setup?hl=es#add-sdk-and-initialize>

³⁰ <https://firebase.google.com/docs/unity/setup?hl=es-419>

Desde la interfaz del juego se añadirá un nuevo botón denominado “Upload Room”, el cual abrirá una ventana que permitirá al usuario autenticarse o en el caso de ser un nuevo usuario registrarse utilizando un correo electrónico y una contraseña, así como un nombre de usuario. Una vez completado el registro, el usuario podrá subir a la página web la habitación que ha terminado de diseñar.

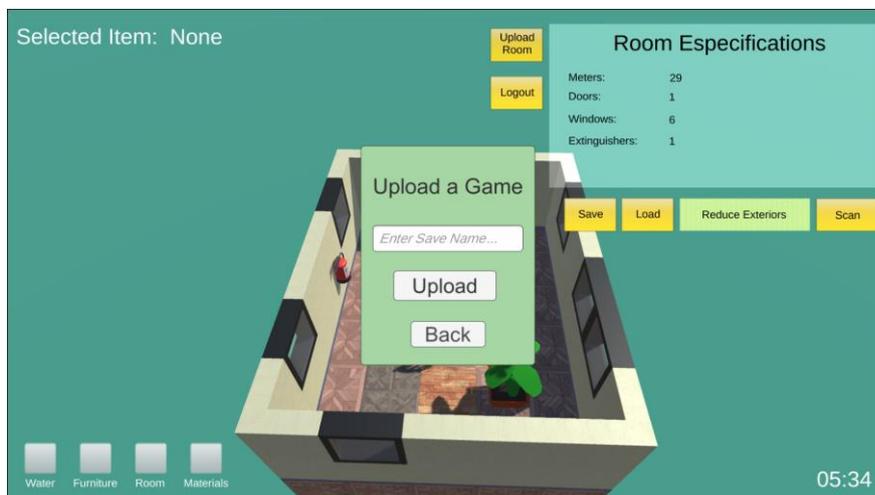


Figura 57. Interfaz para subir un juego a Firebase.

Se enviarán un archivo “. dataRoom” con la información de la habitación diseñada para que sea guardando en la Cloud Firestore de Firebase, además de la imagen de la habitación, la cual será utilizada como previsualización en la página web.

Desde la página web se podrá descargar cualquiera de las habitaciones que se encuentren en la lista dinámica que se ha implementado. Para cada elemento se podrá consultar los detalles y estadísticas de la habitación como el número de metros que ocupa o el número de extintores que tiene.

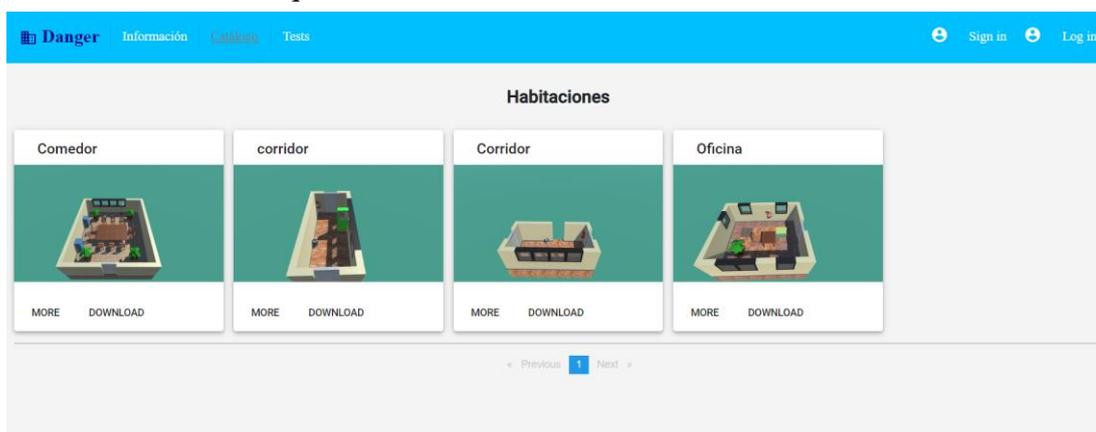


Figura 58. Apartado de Catálogo de la página web

Se podrá descargar cualquiera de las habitaciones que se hayan subido a la página web, mover el archivo a “AppData\LocalLow\DANGER\DANGER” y poder abrir el archivo desde la lista de habitaciones almacenadas en memoria.

6. Conclusiones

6.1 Comparación con la última versión(changelog)

Los apartados que se han mejorado respecto a la versión anterior de juego son los siguientes:

- Se ha añadido un sistema de guardado para el modo Construcción
- Se ha mejorado la estética de los diferentes modos de juego
- Se han añadido nuevos elementos para la creación de habitaciones en el modo Construcción
- Ahora es posible previsualizar los objetos antes de colocarlos en la escena del modo Construcción
- Se han añadido nuevas acciones que podrán realizar los personajes: usar los extintores y moverse a nivel del suelo.
- Se han añadido nuevos estados. Los personajes podrán entrar en pánico cuando su estrés llegue al límite y morirán cuando su vida llegue a 0.
- Ahora es posible configurar los parámetros del generador procedural del modo Escape, como el número de habitaciones y NPC que aparecerán en el nivel.
- Se ha diseñado un nuevo sistema de generación procedural basados en la utilización de habitaciones prefabricadas para aumentar la variedad y el diseño de las habitaciones.
- Se han añadido extintores, alarmas y puertas al diseño del modo Escape. El jugador y los NPC podrán interactuar con estos elementos en su lucha por escapar del fuego.
- Se ha mejorado la percepción del entorno para los personajes, ahora pueden detectar el fuego, humo y elementos que se encuentren en la escena para interactuar con ellos.
- Se ha mejorado la inteligencia artificial de los NPC en todos los modos de juego aplicando árboles de comportamiento.
- Se ha diseñado un nuevo nivel para el modo Aventura.
- La página web ahora dispone de un catálogo con habitaciones creadas por los usuarios. Estas se podrán descargar.
- Se ha rediseñado el apartado Test de la página web, ahora presenta un cuestionario con una batería de 10 preguntas.

6.1.1 Relación con los estudios cursados

En la carrera de Ingeniería informática tuve la oportunidad de cursar la asignatura “Desarrollo de videojuegos en 3D”, aprendí los conceptos básicos de desarrollo de videojuegos usando Unity. Siendo el principal motivo que me impulsó a escoger esta temática para desarrollar mi trabajo de fin de grado.

Además, durante mis estudios en el grado de Ingeniería informática he aprendido distintos métodos para la implementación de la inteligencia artificial, en asignaturas “Sistemas Inteligentes” y “Agentes inteligentes” se nos encargó desarrollar una inteligencia artificial para los NPC en un juego de capturar la bandera donde estos podían actuar como defensa o ataque. Debido a esto me entró curiosidad por aprender cómo se implementa la inteligencia artificial en el diseño de los videojuegos.

Para el desarrollo de la web me fue necesario aprender el funcionamiento del framework Angular. Debido el lenguaje de programación para diseñar páginas web se basan en TypeScript, HTML y CSS, tenía algunas de ese conocimiento gracias a la asignatura “Tecnología de sistemas de información en la red”.

Además, en la asignatura “Desarrollo de aplicaciones para dispositivos móviles” tuve la suerte de poder tener algo de contacto con la SDK de Firebase, aprendiendo a comunicar, transmitir información entre aplicaciones móviles, esto me fue de especial utilidad para tener una base sobre como implementarlo en Angular y Unity.

El desarrollo de un videojuego es una tarea muy compleja que requiere el trabajo de un equipo multidisciplinario de personas con habilidades en diversas áreas. Ha sido muy enriquecedor aprender el proceso de creación de un videojuego, desde la forma en que se generan las ideas, hasta la forma en que se programa los objetivos a conseguir.

6.1.2 Trabajos futuros.

El videojuego está en su mayor parte completo, pero todavía quedan algunos apartados en los que se puede mejorar

La jugabilidad. En esta versión se han implementado mejoras en este aspecto mediante la introducción de nuevas mecánicas que los jugadores podrán llevar a cabo, pero sigue volviéndose monótono y repetitivo al cabo de unas cuantas partidas, es necesario seguir introduciendo nuevas mecánicas y niveles para mantener el interés del jugador.

La portabilidad para dispositivos móviles, uno de los objetivos planteados en el diseño del videojuego poder ejecutar el videojuego en un dispositivo móvil, para conseguir este objetivo será necesario adaptar las interfaces y los controles a las pantallas táctiles de los dispositivos móviles.

La recolección de datos, los datos que se pueden analizar actualmente provienen del modo construcción, se puede establecer un sistema para recopilar datos de la partida en el modo aventura, de esta forma se podrán analizar las partidas jugadas y extraer estadísticas útiles para mejorar el videojuego.



7. Agradecimientos

Principalmente quiero agradecer a Ramón y Teresa por su ayuda en la planificación, información y organización en este Trabajo de Fin de Grado.

A mi madre Eva, mi padre Jesús y mi hermano Ángel que han estado apoyándome a lo largo de toda mi carrera y animándome a seguir adelante.

A mis amigos José Manuel y Daniel por ayudarme a decidir el tema de mi proyecto, así como enseñarme la programación en TypeScript y Angular.

Por último, quiero agradecer a todos los profesores que me han enseñado a lo largo de la carrera, he ganado valiosos conocimientos los cuales me serán de gran utilidad lo largo de mi vida.

8. Bibliografía

- ARDILA-MUÑOZ, J.Y., 2019. Supuestos teóricos para la gamificación de la educación superior. *Magis. Revista Internacional de Investigación en Educación* [en línea], vol. 12, no. 24, pp. 71-84. ISSN 2027-1174. Disponible en: <https://www.redalyc.org/articulo.oa?id=281060624006>.
- ASOCIACIÓN ESPAÑOLA DE VIDEOJUEGOS, 2021. La industria del videojuego en España en 2021 - Anuario 2021. [en línea]. S.l.: [Consulta: 17 junio 2022]. Disponible en: http://www.aevi.org.es/web/wp-content/uploads/2022/04/AEVI_Anuario_2021_FINAL.pdf.
- CORNELLÀ, P., MERITXELL, E. y DAVID, B., 2020. Gamificación y aprendizaje basado en juegos. *Enseñanza de las Ciencias de la Tierra* [en línea], vol. 28, no. 1 SE-. Disponible en: <https://raco.cat/index.php/ECT/article/view/372920>.
- CRAWFORD, C., PEABODY, S., ART, T., WEB, W.W. y LOPER, D., 2003. *The Art of Computer Game Design* [en línea]. S.l.: Osborne/McGraw-Hill. ISBN 0881341177. Disponible en: https://www.digitpress.com/library/books/book_art_of_computer_game_design.pdf.
- FUNDACIÓN MAPFRE y APTB, 2021. Víctimas de incendios en España en 2020. [en línea]. S.l.: [Consulta: 17 junio 2022]. Disponible en: <https://documentacion.fundacionmapfre.org/documentacion/publico/es/media/grup/1164.do>.
- Global Gamers by Region - DFC Intelligence. [en línea], 2021. [Consulta: 17 junio 2022]. Disponible en: <https://www.dfciint.com/global-gamers-by-region/>.
- HUIZINGA, J., 2012. *Homo ludens*. book. [3ª ed.]. Madrid: Alianza. El libro de bolsillo Alianza ; Humanidades ; 22. ISBN 9788420608532.
- PÉREZ-LÓPEZ, I.J., 2018. La docencia es un juego donde gana el que más disfruta. *Habilidad Motriz*, pp. 2-3.
- PROGRAM ACE, 2021. Unity vs. Unreal: What to Choose for Your Project? [en línea]. Disponible en: <https://program-ace.com/blog/unity-vs-unreal/>.
- ROMERO SANDÍ, H. y ROJAS RAMÍREZ, E., 2013. La Gamificación como participante en el desarrollo del B-learning: Su percepción en la Universidad Nacional, Sede Regional Brunca. *Innovation in Engineering, Technology and Education for Competitiveness and Prosperity: Proceedings of the 11th Latin American and Caribbean Conference for Engineering and Technology* [en línea], pp. 10. [Consulta: 18 junio 2022]. Disponible en: <http://www.laccei.org/LACCEI2013-Cancun/RefereedPapers/RP118.pdf>.
- RUTH S. CONTRERAS ESPINOSA Y JOSE LUIS EGUIA (EDS.), 2016. *Gamificación en aulas universitarias* [en línea]. S.l.: s.n. [Consulta: 25 junio 2022]. ISBN 978-84-944171-6-0. Disponible en: <https://core.ac.uk/download/pdf/78545392.pdf>.
- SHUTE, V.J. y KE, F., 2012. Games, learning, and assessment. *Assessment in Game-Based Learning: Foundations, Innovations, and Perspectives*. S.l.: s.n., pp. 43-58. ISBN 9781461435464.
- UNICEF, 2018. Aprendizaje a través del juego. *The LEGO Foundation* [en línea], pp. 84. [Consulta: 23 junio 2022]. Disponible en: <https://www.unicef.org/sites/default/files/2019-01/UNICEF-Lego-Foundation->



Aprendizaje-a-traves-del-juego.pdf.

YU-KAI CHOU, 2019. The Octalysis Framework for Gamification & Behavioral Design. [en línea]. [Consulta: 24 junio 2022]. Disponible en: <https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>.

Anexo I: Glosario

Assets: Son objetos de Unity que podemos utilizar en nuestros juegos, representan distintos tipos de recursos dentro del programa, desde modelos 3D, como archivos de sonido, elementos de la interfaz de usuario ... etc.

Avatares: Son personajes en el videojuego que pueden ser controlados por el jugador.

Escena: Una escena en Unity es un asset que contiene parte del videojuego o la aplicación que se esté desarrollando, se puede crear cualquier número de escena en el juego y normalmente cada escena representa un nivel o entorno dentro del videojuego.

GameObject: Son objetos en Unity que representan personajes, obstáculos, etc. Cualquier elemento que se encuentre en una escena se considera un GameObject.

UI: La interfaz de usuario permite al videojuego interactuar con el usuario. En la mayoría de los casos estos elementos son paneles o elementos interactivos como botones o campos en blanco con datos a rellenar por parte del usuario.

Niveles: Representa un espacio presentado a jugador con el objetivo que resuelva un desafío, el cual irá escalando en dificultad en sucesivos niveles.

Prefab: Un Prefab es un tipo de datos de Unity que permite crear, configurar y guardar un GameObject y sus propiedades, actúa como una plantilla a partir de la cual se pueden crear nuevas instancias del objeto en la escena.

Raycast: Un Raycast en Unity es una función de física que permite proyectar un rayo en la escena

Collider: Es un componente que permite envolver a un GameObject con el objetivo de detectar las colisiones entre diferentes objetos dentro de un nivel. Generalmente representan figuras geométricas como una esfera, un cilindro o un cubo.

Usuario: Son las personas que adquieren la copia de software, también denominamos jugadores en el caso de videojuego.

Navmesh: Es una representación de un espacio de navegación utilizado por la inteligencia artificial. Se puede usar para encontrar caminos a través de un espacio para que un agente pueda moverse.

Hierarchy: Este componente contiene todos los GameObject en la escena actual. Se ordenan utilizando estructura jerárquica donde se pueden establecer relaciones de Padres e hijos entre los objetos, estando los hijos contenidos dentro de los objetos padres.



Anexo II: ODS

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.		X		
ODS 4. Educación de calidad.	X			
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.	X			
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.			X	
ODS 17. Alianzas para lograr objetivos.				X

Reflexión

Los incendios han supuesto un tema serio y difícil de tratar. Un siniestro relacionado con un incendio mal gestionado puede llegar a provocar grandes daños tanto monetarios, como humanos y ambientales.

Las herramientas orientadas a fines educativos como DANGER ayudan a las personas a adquirir una formación sobre los métodos de prevención de incendios, son herramientas útiles en aquellos lugares en los que el riesgo incendio es elevado o en lugares donde un siniestro de este tipo podría ocasionar muchas pérdidas.

Los objetivos de desarrollo sostenible (ODS) con los que este proyecto está relacionado son:

Educación de calidad: Los videojuegos como método de enseñanza son una de las formas educativas que más llaman la atención a niños y adolescentes. DANGER aprovecha las innovaciones tecnológicas y el uso de los videojuegos para llegar a ellos de una forma divertida y efectiva, concienciándolos y enseñándoles medidas de prevención de incendios y a tomar decisiones correctas en caso de que se produzca tal incidente.

Acciones por el clima: Los incendios urbanos son un problema porque causan daños materiales que deberán ser, aunque no afecten significativamente a la liberación de CO₂ en la atmósfera, los incendios forestales suponen un problema mucho mayor, ya que destruyen áreas de vegetación y liberan grandes cantidades de CO₂ a la atmósfera. Sensibilizar a la gente sobre estos problemas ayudará a mitigar el cambio climático.

Ciudades y comunidades sostenibles: El juego tendrá un impacto positivo en la reducción de muertes y daños causados por estos, desastres, ya que los jugadores podrán aprender técnicas de prevención de incendios. Se espera que el juego contribuya a mejorar la calidad del aire en las ciudades, al reducir la cantidad de humos emitidos por incendios, además de reducir los gastos públicos destinados a la reconstrucción de zonas afectadas por estos desastres.

Salud bienestar: El juego DANGER contribuye a la reducción de incendios y enseña a la gente cómo autoprotegerse en el caso de emergencia, lo que a su vez fortalece las capacidades de alerta temprana personal, reduce el riesgo y gestiona el riesgo de incendio a nivel internacional. Como resultado, ayudará a reducir sustancialmente el número de muertes y la contaminación del aire, el agua y el suelo.

Anexo III: Documento del diseño del juego

SECCIÓN 1: DATOS PRINCIPALES

- **Título:** “D.A.N.G.E.R – Disaster Prevention”
- **Concepto abstracto:** *Danger* es un juego con propósito pedagógico con el que aprender normativas y procedimientos aplicables a la prevención de incendios.
- **Tema:** Sociedad actual, gestión de emergencia, simulación.
- **Género:** Construcción y estrategia en tiempo real.
- **Plataforma:** Windows/Android/IOS.
- **Mercado:** Todos los públicos.
- **PEGI:** Para 7 años o más. Intentamos hacer para que el juego sea aceptable para todos los rangos de edad, aun así, pueda haber unas escenas que muestra crueldad de los incendios, y no son muy apropiados para niños menor que 7 años.



FIGURA GDD 1. PEGI 7

Equipo:

1.1 **Desarrolladores:**

1.1.1 Adrián Sánchez Lavarias (email: adsanla@inf.upv.es)

1.1.2 Pablo Querol Ballester (email: pabqueba@inf.upv.es)

1.1.3 Xinyu Jiang (email: xinjia@inf.upv.es)

1.1.4 Luis Alberto Alvarez Zavaleta (email: lualza1@etsii.upv.es)

1.2 **Consultores técnicos:**

1.2.1 Yennifer Ramirez Lopez (email: yenralo@posgrado.upv.es)

SECCION 2: Pitch doc

Resumen:

En juego puedes elegir entre actuar como el diseñador de un edificio (modo construcción) o un público que accidentalmente queda atrapado en un incendio (modo estrategia o aventura).

Por lo tanto, cuando juegas como un diseñador, podrás construir salas y ponerlas a prueba para comprobar si cumplen con la normativa.

Y si quedas atrapado en un edificio en llamas, hay que guiarlas para que logren escapar antes de que les alcance el fuego.

Gameplay:

Mezclará los géneros de construcción, puzle y estrategia en tiempo real en tercera persona, enfatizando en la toma de decisiones a lo largo del videojuego

En modo de construcción, tendrás que aprovechar todos los bloques y objetos que te ofrecen, arrastrarlas (Con ratón o pantalla táctil) para crear viviendas u oficinas que cumplen la normativa de prevención de incendio.

En modo de estrategia de tiempo real, también tienes que usar ratón o pantalla táctil para indicar el destino de los personajes para que pueda salir de la situación de emergencia. En este modo también encontrarán una serie de obstáculos como puzles o preguntas que hay que resolver para poder continuar con la evacuación.

Para el modo de estrategia de tiempo real se dispone dos formas de representar la escena:

- Escena generada de forma procedural: Escenas aleatorias creado con algoritmos, por lo tanto, cada vez que entra a jugar es un nuevo nivel, para mantener intereses de los jugadores (modo escape).
- Escena construida manualmente: Niveles prediseñados, servirán tanto para niveles introductorios como niveles finamente diseñados para garantizar dificultades razonables (modo aventura).

Finalidad del juego:

El objetivo es hacer un juego:

- Interactivo en 3D, con el fin de enseñar la normativa y metodología en caso de incendio. Para ello se han creado 3 modos de juego de los géneros construcción con puzles y estrategia en tiempo real. Lo que permite desarrollar la creatividad del usuario y colocarlo en una simulación de una situación real.



- Diferenciador, la dualidad de modos de juego que desarrollan la creatividad, la capacidad de estrategia y reacción del jugador. Ambos modos de juegos, además, fomentan la *rejugabilidad* del título.

Estética:

El juego será en 3D, el modo construcción será en primera persona, mientras que el modo construcción y aventura es de tercera persona. Los tres modos del juego presentarán la misma estética.

Dado que la escena del juego es el mundo real y tiene un tema serio, por lo tanto, los personajes, las escenas y las animaciones del juego será realista acortando todos los detalles con el mundo real.



FIGURA GDD 2. Imagen conceptual, fuente GTA 5

SECCIÓN 3: Análisis competitivo

Referentes

Nuestras competencias serán todas las metodologías para realizar la formación de prevención de riesgo contra los incendios.

A grandes rasgos, está:

- Fuera de contexto del juego: simulacro de incendios.
- Juegos/Juegos de mesas.
- Videojuegos.
- Producto de realidad virtual.

Se procede a realizar análisis DAFO para hacer una comparativa con otro producto:

- **Simulacro de incendios.**

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - El usuario no puede revisar su equipamiento real contra incendio. - A veces es menos eficaz que un simulacro.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades</p> <ul style="list-style-type: none"> - Puede jugar en cualquier momento. - Puede entrenar en diferentes escenas. - Puede enfrentar casos imprevistos.

FIGURA GDD 3. Análisis DAFO de *Danger* con simulacro (Desde la perspectiva de *Danger*).

- **Juegos/Juegos de mesas:** *The Great Fire of London 1666.*

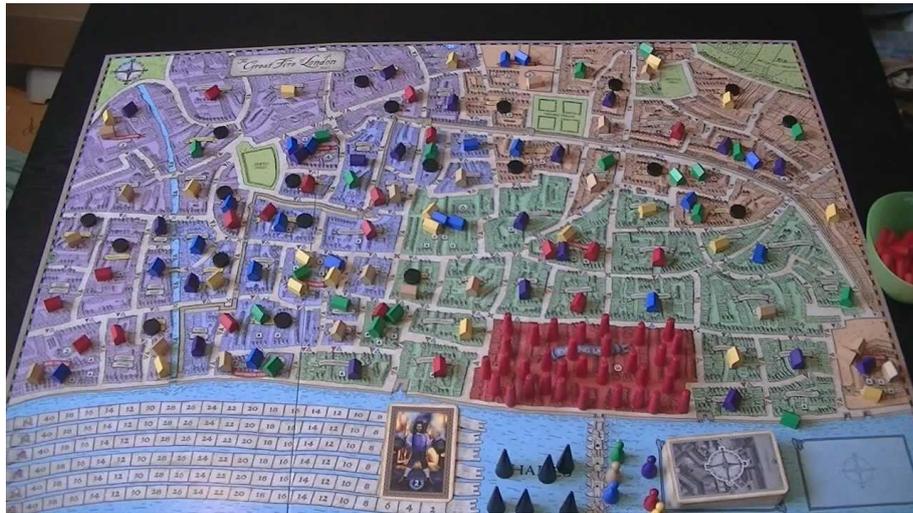


Figura GDD 4. The Great Fire of London 1666.

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - Es menos divertido, ya que normalmente se juega solo.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades</p> <ul style="list-style-type: none"> - Danger está basado en sociedad actual. - Normalmente enfrentas con problemas reales, mientras que enfrentar la situación de <i>The Great Fire of London 1666</i> es muy poco probable.

FIGURA GDD 5. Análisis DAFO de *Danger* con *The Great Fire of London 1666* (Desde la perspectiva de *Danger*).

- **Videojuegos:** *Flame Over.*

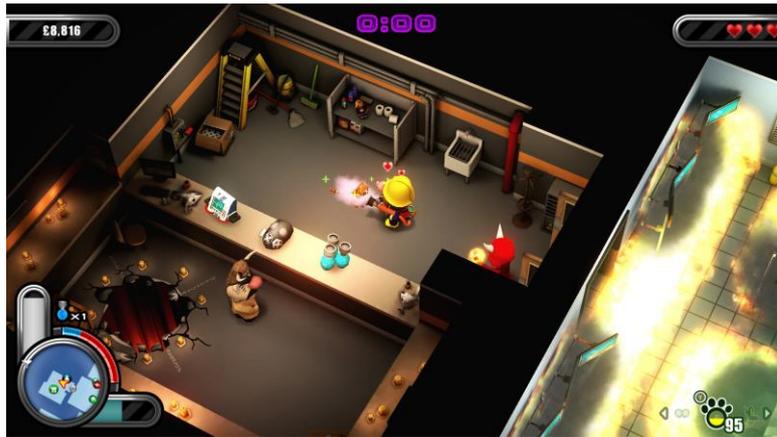


Figura GDD 6. Flame Over.

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - En Danger no puedes actuar como un bombero. - La gráfica es menos llamativa.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades</p> <ul style="list-style-type: none"> - Dispone gráfica más realista. - Dispone un sistema de preguntas y respuestas para consolidar el conocimiento.

FIGURA GDD 7. Análisis DAFO de *Danger* con *Flame Over* (Desde la perspectiva de *Danger*).

- **Producto de realidad virtual: Ludus.**



Figura GDD 8. Ludus.

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - <i>Danger</i> es menos inmersivo comparando un producto de realidad virtual. - Es menos interactivo comparando un producto de realidad virtual.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades</p> <ul style="list-style-type: none"> - Dispone niveles que se genera <i>proceduralmente</i>, mientras que Ludus solamente dispone un nivel. - Es más barato, no requiere cascos de realidad virtual.

FIGURA GDD 9. Análisis DAFO de *Danger* con *Ludus* (Desde la perspectiva de *Danger*).

SECCIÓN 4: Estructura narrativa, personajes

Estructura narrativa:

Los modos de construcción y escape son modos sin narrativas, el jugador simplemente puede elegir el modo y empezar a disfrutar el juego.

En el modo de aventura, tampoco hay una historia principal, pero sí que hay descripciones de lo que está sucediendo en la escena y también se explica cuáles son los objetivos de los jugadores.

Personajes:

<u>Modo</u>	<u>Nombre</u>	<u>Descripción</u>	<u>Foto</u>
Modo construcción	Protagonista	El protagonista del modo de construcción.	No visible dentro del juego
Modo escape	Protagonista	El protagonista del modo de aventura.	
Modo aventura	Protagonista – Nivel 1 y Nivel 2	Un oficinista queda atrapado en un edificio, tienes que ayudarlo a escapar.	
Modo aventura	Personajes no controlables – Nivel 1 y Nivel 2	Compañeros de trabajo del protagonista, puedes decidir si quieres (o si eres capaz) de salvarles.	

Todos los modos	Puede aparecer más personajes		
-----------------	-------------------------------	--	--

SECCIÓN 5: Escenas, Niveles

Escenas

El juego comienza con una Cinemática introductoria, se puede esperar o saltar con la tecla espacio o escape. Sirve como primer contacto con el jugador, por lo que hemos creado una escena móvil con música para introducir al jugador a nuestro juego.

Una vez finalizado el *Splash* encontraremos en el menú principal, desde aquí podemos acceder a las demás escenas.

Desde menú principal puedes acceder a diferentes opciones: créditos, configuración, modo construcción, modo estrategia, modo aventura.

Configuración: Esta escena servirá para que el jugador pueda cambiar entre pantalla completa o ventana, modificar el volumen, los gráficos del videojuego o la resolución a la que se muestra. Con el botón “back” volveremos al menú principal.

Créditos: En esta escena hemos colocado las distintas tareas de las que hemos encargado cada uno en el proyecto en un fondo negro con música. Esta escena conecta con el *Splash* para que la navegación sea completa.

Modo construcción: Este es nuestro primer modo *jugable*, mantiene la música de las anteriores escenas, se compone por un entorno interactivo en el que construir salas de cualquier forma y tamaño.

Modo estrategia: Para nuestro segundo modo, contamos con un número de salas generadas de forma procedural, donde encontraremos un modo de juego de estrategia en tiempo real.

Modo aventura: El tercer modo, es parecido como modo de estrategia, pero con niveles predefinidos.

Este es un diagrama de flujo para ver relaciones entre las escenas:

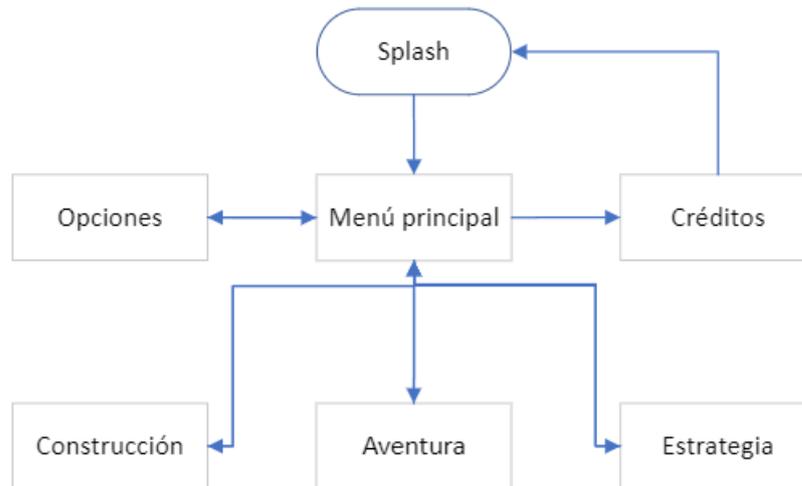


FIGURA GDD 10. Diagrama de flujo.

Modo construcción

En el modo Construcción encontramos un nivel con un solo bloque base desde donde el jugador comenzará a crear el conjunto de salas deseado. Estas salas tendrán unos requisitos dependiendo del tamaño total de las mismas que deben cumplirse para finalizar el nivel.

Elementos jugables:

- Construcción de escena: Elegir bloque y construir una habitación de tu imaginación.

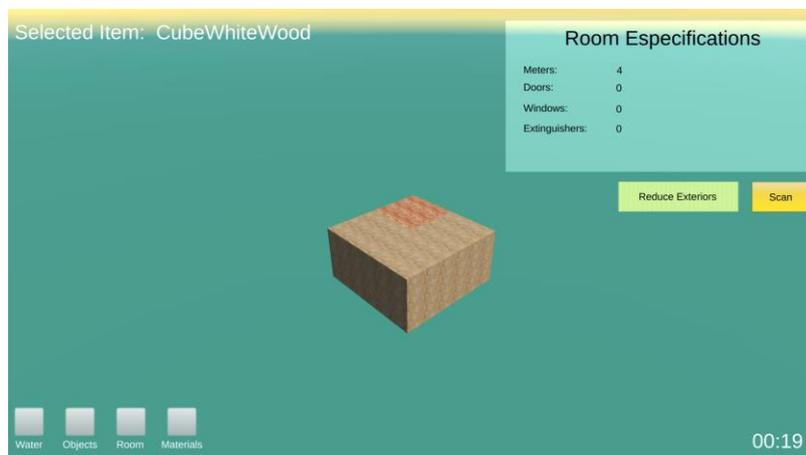


FIGURA GDD 11. Una habitación en construcción.

- Verificación de resultado: Con el botón “Scan” compruebas si tu habitación cumple requisitos.



FIGURA GDD 12. Comprobando si la habitación cumple requisitos con el botón Scan.

Nivel:

El juego ofrece un “sandbox” que permite a los jugadores construir cualquier escena y luego comprobar y cumple la normativa, por lo tanto, no hay niveles explícitos.

Modo estrategia

Los niveles del modo “escape” se generan de forma procedural, por lo que cada vez que se ejecuta el juego es diferente. Sin embargo, todos comparten que son salas conexas entre sí con forma rectangular poblada con mobiliario de oficina.

Elementos jugables:

- Control de personajes: Controla tu personaje para escapar de aquí.

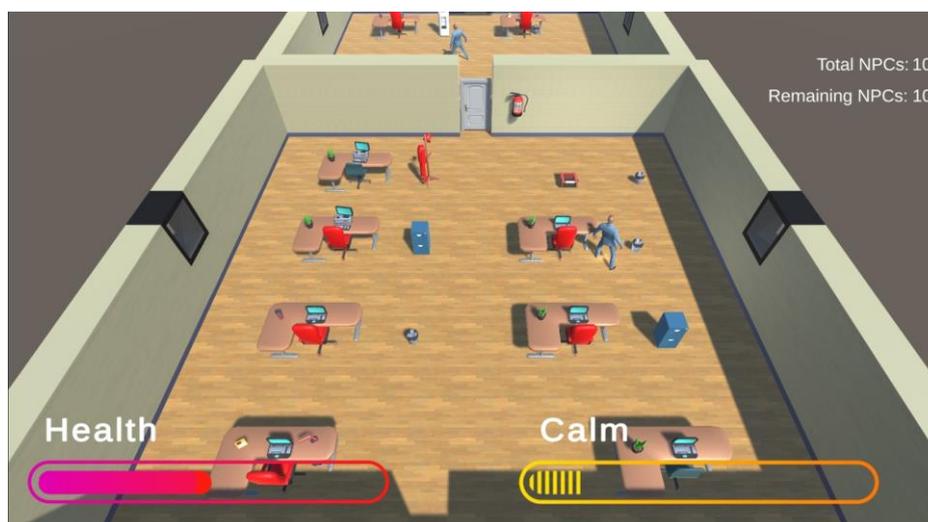


FIGURA GDD 13. Personaje que se encuentra en una oficina.

Nivel:

Gracias al algoritmo de generar escenas automáticamente, el juego existe infinitas posibles escenas, por lo tanto, existe infinitos niveles.

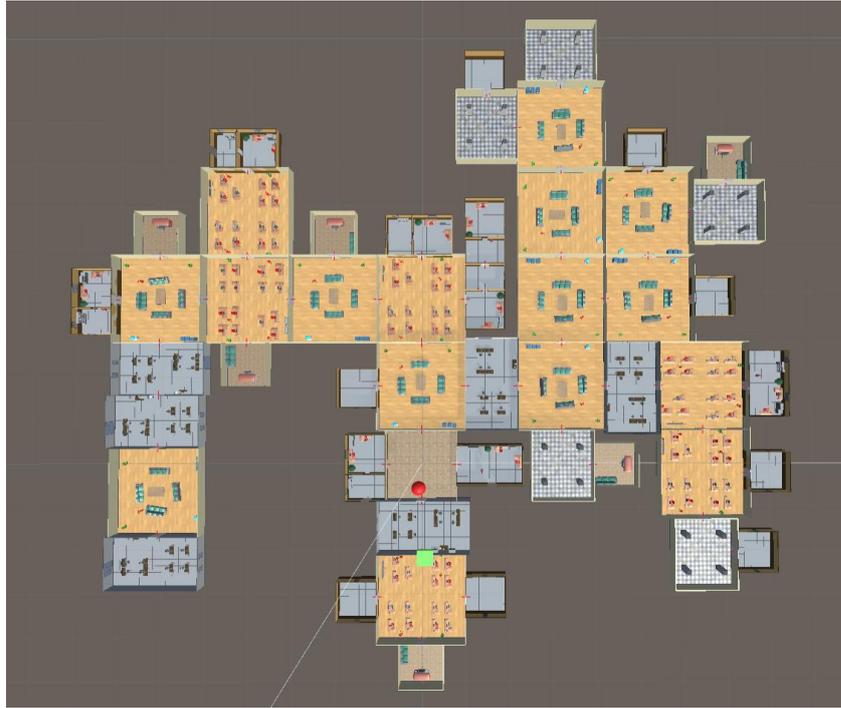


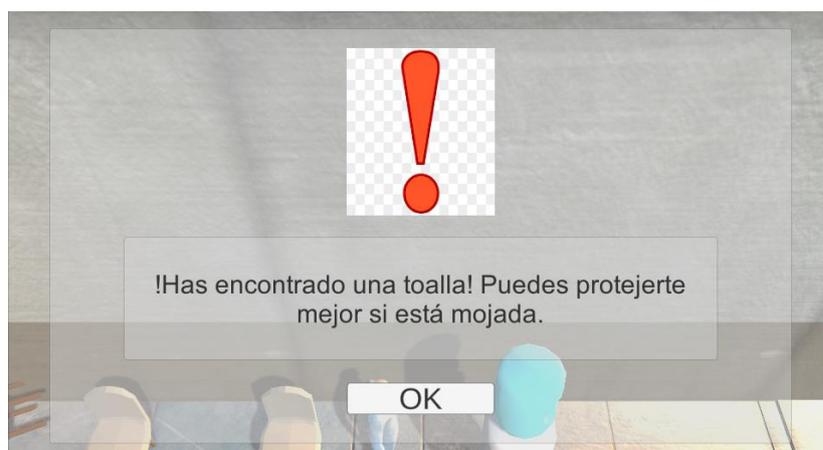
FIGURA GDD 14. Diseño generado de forma aleatoria

Modo aventura

También con modo Aventura, es similar en cuanto al diseño del modo Escape, pero las salas se construyen de forma manual, de esa forma, podemos ofrecer a los usuarios unos niveles bien diseñados y dificultades controlables.

Elementos jugables:

- Interacción con el entorno: ¡Interacciona y colecciona los objetos, te puede salvar la vida!



- Gestión de situación de emergencia: Aprovecha todos los elementos del entorno y tu inventario para salir de la emergencia.



FIGURA GDD 15. Personaje rodeado por la llama.

- Contestar las preguntas: Ante la situación de emergencia, debes tomar decisión correcta para poder avanzar.



Nivel 1:

El jugador comienza desde la habitación número uno, en esta habitación vamos a encontrar dos puertas, una se encuentra bloqueada por el fuego y la otra no. Opcionalmente, el jugador puede pulsar alarma para recibir una bonificación.

La habitación dos también se encuentra con dos puertas, y ambas se pueden abrir. Pero fijamos que hay una señal de salida con flecha, si el jugador abre la puerta que no está apuntada por la señal, recibirá una penalización.

La tercera habitación es una zona con humo oscuro, si el jugador utiliza el pañuelo húmedo para cubrir vías respiratorias recibirá otra recompensa, y si quiere seguir avanzando, tiene que contestar una pregunta sobre cómo pasar el recinto con presencia de humo espeso.

En la cuarta habitación otra vez encontramos con dos puertas, una de ella está caliente y no se debe abrir en un incendio, si el jugador intentar abrir, le penalizamos.

La habitación quinta y sexta encontramos con puertas bloqueadas y personajes no controlables atrapados en esta habitación, el jugador debe responder preguntas para poder salvar a todos ellos y abrir las puertas.

En la séptima habitación se encuentra con un extintor y una única puerta bloqueada por el fuego. Para poder continuar, el jugador debe contestar la pregunta de cómo usar el extintor para poder apagar el fuego con éxito.

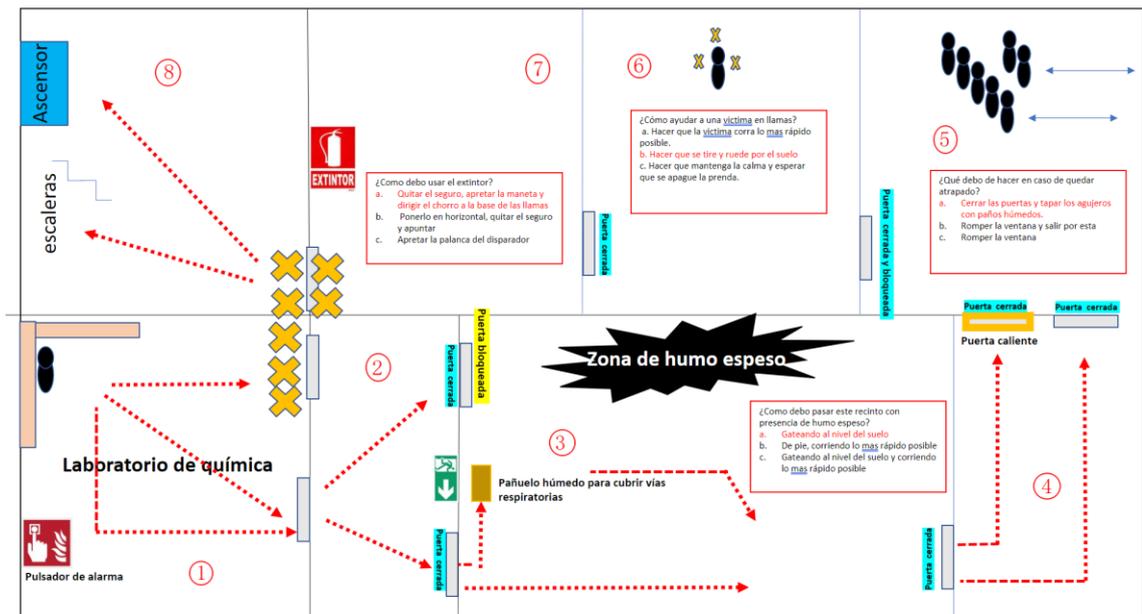


FIGURA GDD 16. Esquema conceptual del nivel 1.

Nivel 2:

La primera habitación será en lugar donde el jugador aparecerá, el jugador deberá apagar el fuego que se ha generado en la escena utilizando el extintor que se ha ubicado a la salida de la habitación. Poco después de que se apague el fuego empezará a crecer con más fuerza y el jugador tendrá que evacuar de la habitación.

En la segunda habitación del jugador deberá avisar a sus compañeros sobre el incendio, estos acompañarán al jugador hasta la salida. El jugador deberá acercarse a ellos para hablar.

En la tercera habitación el jugador tendrá que rescatar al NPC que ha sido rodeado por las llamas, teniendo que usar la carga restante del extintor, y en caso de no ser suficiente podrá tomar el que se encuentra en el pasillo de la habitación.

En la cuarta habitación el jugador podrá encontrar un teléfono con el que podrá avisar a los bomberos para que acudan al edificio.

La quinta y sexta habitación contendrán más NPC con los que el jugador podrá interactuar para ayudarles a dirigirse a la salida.

Además, cerca de la quinta habitación se encontrará la alarma de incendios, que podrá ser activada.

La séptima habitación será una habitación en llamas, el jugador deberá estar atento a las señales que se proporcionan por el mapa para evitar entrar en ella y dirigirse a la salida a través de la octava habitación.

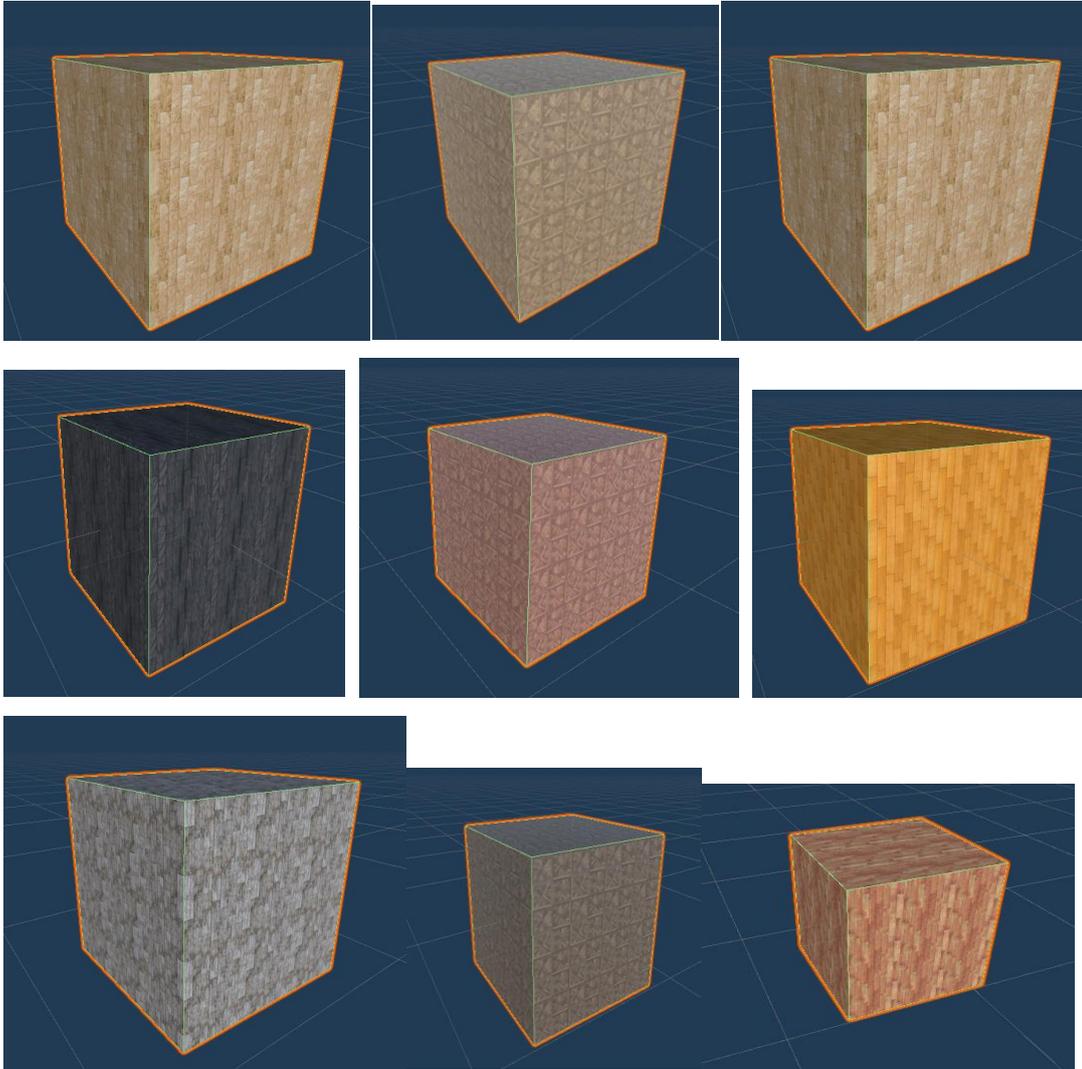
En la octava habitación el jugador deberá pasar agachado por las cantidades de humo que se han generado desde la séptima habitación, en caso de que el jugador pase sin estar agachado la vida de jugador irá decayendo rápidamente.

En la novena habitación el jugador deberá apagar las llamas que se encuentran cerca de la salida, se ha colocado un extintor adicional en la puerta entre la octava y la novena habitación para que el jugador pueda completar con éxito el nivel.

SECCIÓN 6 Assets:

Modo construcción:

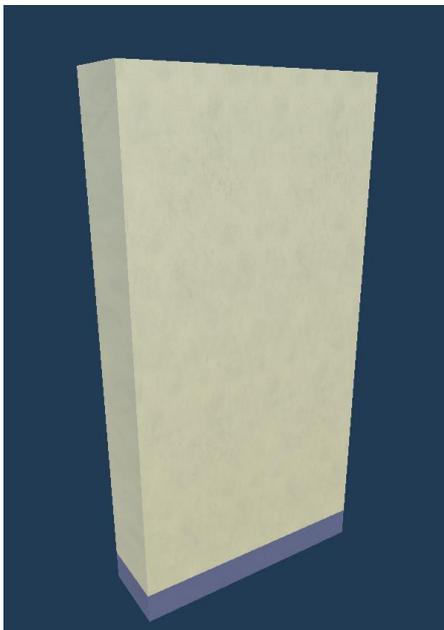
- **Bloques:** Sirve para colocar en el suelo de las habitaciones, la diferencia no sólo está en los colores, también influirá cómo propagar el fuego.



- **Puertas:** Sirve para parar el fuego de manera débil, debe tener al menos una para ser reconocido como una habitación.



- **Muro:** Sirve para parar el fuego, debe rodear sobre bloques de suelo para ser reconocido como una habitación. También sirve para colgar los extintores.



- **Ventana:** Para ventilar, a la hora de construir una habitación hay que tener cantidad necesaria para no saltarse de la normativa.



- **Mesa:** Elemento decorativo, puede dejar objetos sobre ella.



- **Extintor:** Para apagar el fuego.



- **Mobiliario:** Para decorar la habitación.



Modo escape y modo aventura

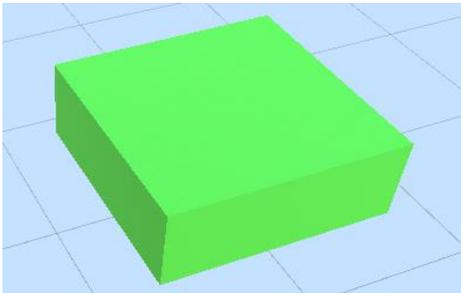
- **Protagonista:** Personaje controlable, hay que guiarle a la salida.



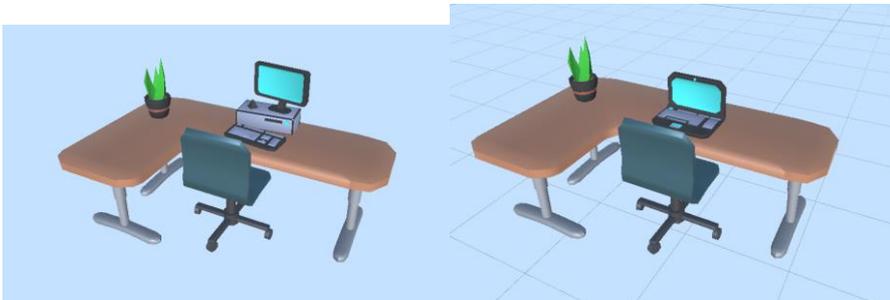
- **Personaje no controlable:** Personaje no controlable, puedes ayudarlo a escapar juntos.



- **Salida:** Tienes que llegar aquí para salir.



- **Mesas:** Decoraciones, interaccionable cuando es necesario.





- **Papelera:** Decoraciones, interaccionable cuando es necesario.



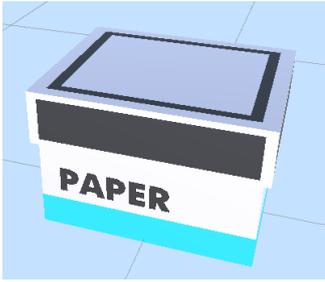
- **Armario:** Decoraciones, interaccionable cuando es necesario.



- **Carro de limpieza:** Decoraciones, interaccionable cuando es necesario.



- **Caja de papeles:** Decoraciones, interaccionable cuando es necesario.



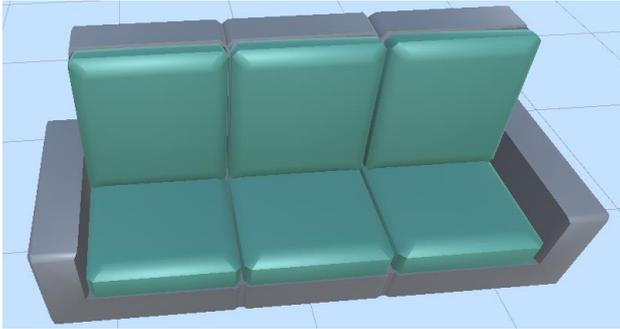
- **Planta:** Decoraciones, interaccionable cuando es necesario.



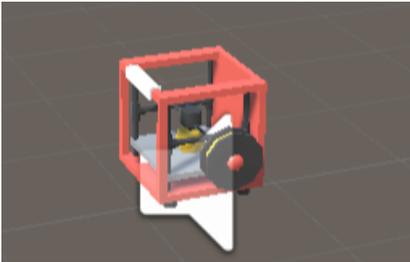
- **Máquina de agua:** Decoraciones, interaccionable cuando es necesario, como para mojar la toalla.



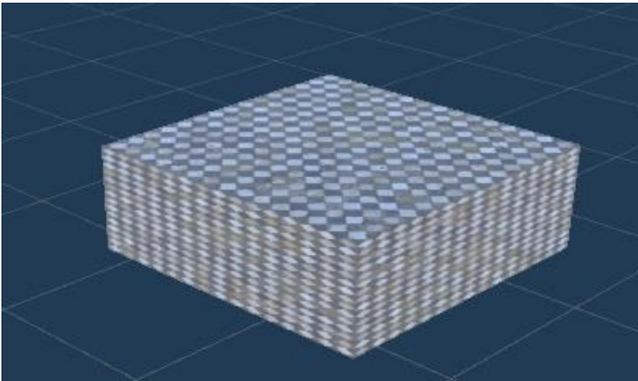
- **Sofá:** Decoraciones, interaccionable cuando es necesario.



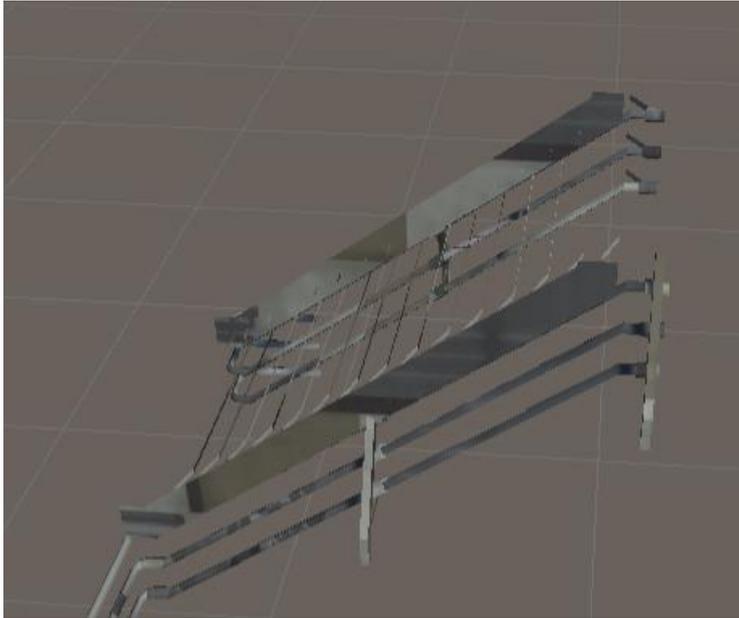
- **Alarma:** Objeto interaccionable, normalmente da *bonus* de tiempos extras.



- **Toalla:** Objeto coleccionable, normalmente tienes que mojar en máquina de agua para poder reducir daño de fuego.



- **Escalera:** Hay que llegar a aquí para ir de una escena a otra.



- **Fuego:** Hace daño cuando estás sobre él y produce humos.



SECCIÓN 7: Controles

En este apartado se tratará de profundizar en detalle en todo lo referido a los controles que usaremos, para así poder realizar las distintas acciones que estarán definidas dentro del videojuego.

Tenemos la intención de lanzar nuestro videojuego en PC y dispositivos móviles, sin embargo, los controles de la versión de móvil no están implementados todavía.

Se añadirá una pequeña introducción en cada modo de juego con el objetivo de que el jugador conozca los controles básicos del modo de juego como el desplazamiento de la cámara, colocar y eliminar objetos dentro de la escena de juego



Teclas por defecto para el modo de juego construcción (Build)

- Desplazamiento de la cámara: teclas WASD.
- Rotación de la cámara: Control Izq + Clic Derecho.
- Zoom de la cámara: Rueda del ratón.
- Colocar un objeto: Clic Derecho.
- Eliminar un objeto: Clic Izquierdo.

Teclas por defecto para el modo Escape y Aventura

- Desplazamiento de la cámara: teclas WASD.
- Mover la cámara: Q y E
- Seleccionar personaje: Clic izquierdo.
- Seleccionar destino: Clic izquierdo.
- Aumentar velocidad del movimiento: *Shift*.
- Agacharse: *Control*
- Abrir/cerrar mapa: tecla M.

- Interacción con entorno: Clic derecho.
- Contestar preguntas: Clic izquierdo.
- Cerrar el juego: Esc.

Menú de Acceso

- Seleccionar destino: Clic izquierdo.
- Interaccionar con el entorno: Clic derecho.
- Seleccionar una respuesta: Clic izquierdo.
- Cerrar el juego: Esc.

SECCIÓN 8: Interfaz

En esta sección concretamos la interfaz del jugador en detalle, especificaremos los distintos menús y cómo moverse entre ellos. Dentro de esta parte encontraremos el menú principal con sus distintas opciones, el HUD y el inventario.

Cámara

Respecto a la cámara tendremos una vista en 3ª persona controlable por el jugador, de manera que el la sitúe en la dirección y ángulo que quiera en cada momento.

Vista estándar:

De manera estándar la cámara se situará en una posición concreta.



Vistas Alternativas

En el modo de estrategia (Escape) disponemos de una vista alternativa en forma de mapa, desde esta vista es la única donde pueden verse la posición actual del jugador en el escenario (punto rojo) y la salida del edificio donde tienen que llevar los personajes para ganar (cuadrados verdes).



Opciones controlables por el Jugador:

Utilizando el ratón el jugador podrá mover la cámara a voluntad.

HUD

Modo de construcción:

Selected Item: CubeRedWood

Room Specifications

Meters: 1
Doors: 0
Windows: 0
Extinguishers: 0

Reduce Exteriors

Scan



Red



Black



Red 2



White



Black 2



Grey



Orange



White 2



Water



Objects

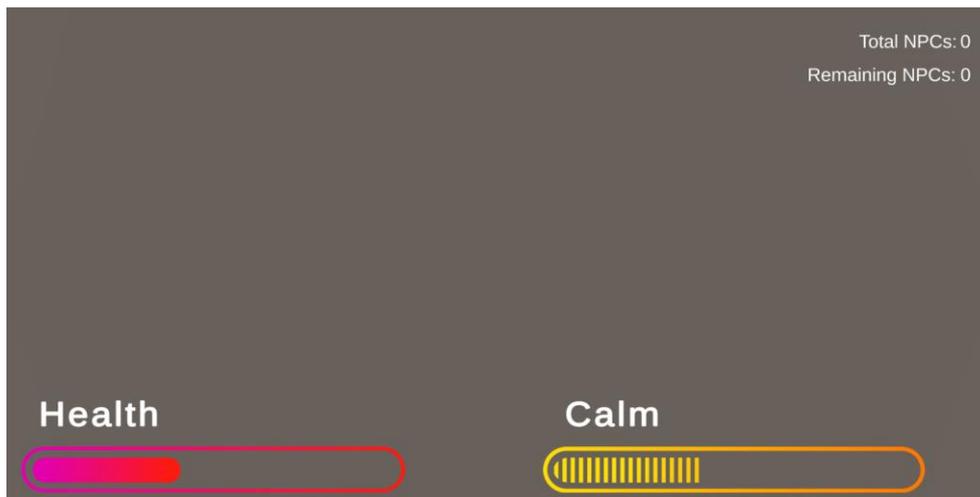


Room



Materials

Modo de estrategia



Modo de Aventura



Información de estado

En la vista del jugador podemos observar los distintos estados e información que le damos al jugador:

Modo escape y modo adventure

- Salud: Visible en la esquina inferior izquierda del HUD y administrada mediante una barra de color gradiente entre rosa y rojo. Cuando el fuego te alcanza al personaje el tamaño de la barra disminuye. Si te quedas sin salud pierdes el juego.
- Estado de ánimo: Visible en la esquina inferior izquierda del HUD y administrada mediante una barra de color gradiente entre amarillo y naranja. Cuando el personaje detecta fuego o humo este nivel aumenta. Cuando el personaje lleva un tiempo sin ver elementos peligrosos comienza a disminuir gradualmente. Cuanto más alto es

este nivel mayor será la velocidad del personaje y menor será el tiempo de espera del personaje antes de pasar al modo autónomo. También aparecerá un texto sobre la barra que se intercambiará entre: “Calm”, “Stress” y “Pánic” según vaya aumentando o disminuyendo el nivel de estrés.

- NPCs totales y restantes: Contadores visibles en la esquina superior derecha con el número de NPCs restantes y totales.

Modo Build

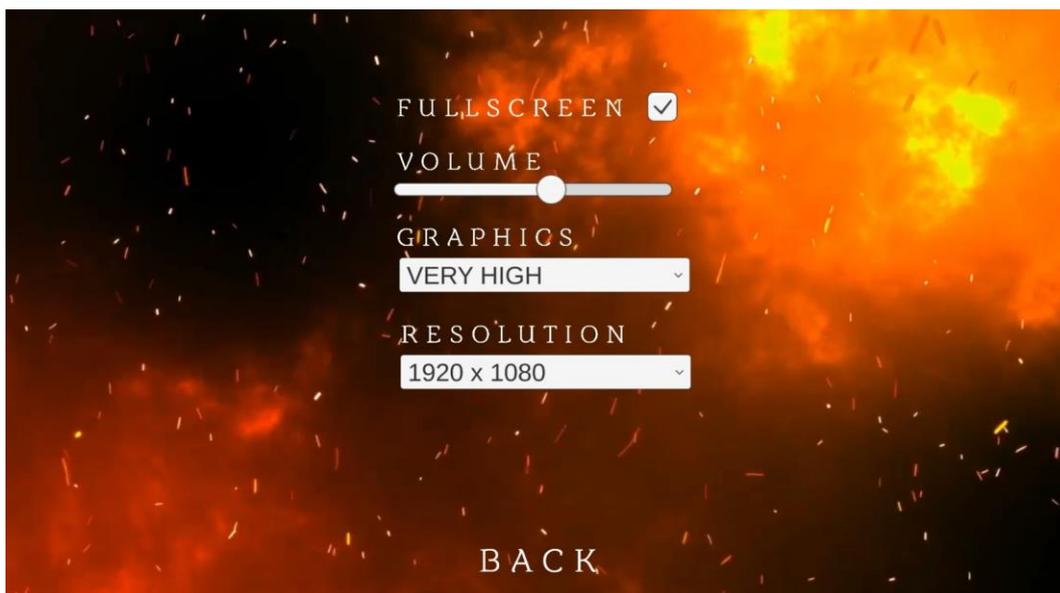
- Selected Item: En la esquina superior izquierda encontraremos este elemento que indicarán qué objeto concreto tiene el jugador seleccionado en cada momento.
- Room Specifications: Este elemento se encuentra en la esquina superior derecha, se compone por un panel con la información relativa a todas las salas creadas por el jugador en la escena.
- Item selectors: Conjunto de botones donde los diferentes objetos se clasifican, al pulsar estos botones abrimos sus páginas correspondientes para poder seleccionar el objeto deseado.

Menús

Menú principal



Menús de opciones



SECCIÓN 9: Herramientas complementarias

Página web

Hay que crear una página web con siguientes finalidades:

- Tener fácil acceso a los distintos enlaces de interés sobre cómo actuar en medidas de emergencia y prevenirlas.
- Registrar los puntos de todas las partidas.
- Repasar las preguntas que aparecían en el juego.

Por lo tanto, que tiene siguiente forma:

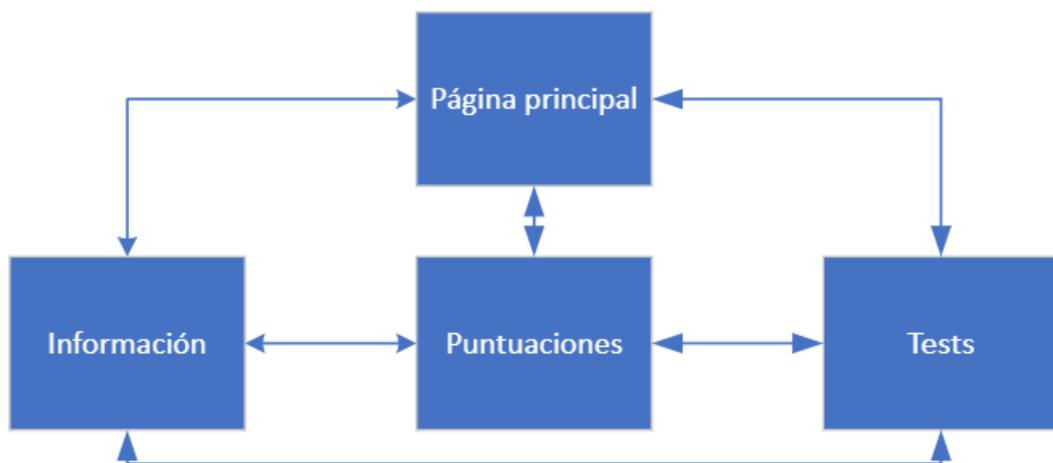


Figura GDD 17. Diagrama de flujo página web.

Y cada una de las funcionalidades corresponden a:

- Página principal: Portal para navegar a otras distintas páginas.
- Información: Acceder distintas fuentes de información sobre qué hacer en caso de incendio.
- Puntuaciones: Ver el historial de los resultados del juego.
- Tests: Consiste en hacer repasos de las preguntas del juego sin tener que abrirlo. También dispone un botón de “Mostrar respuestas” que sirve para ver las respuestas junto con una breve explicación con las referencias para que los jugadores puedan entender mejor porqué la respuesta es así.